



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Realizace pracoviště robotického vidění

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 3906T001 – Mechatronika

*Autor práce:* **Bc. Jakub Hegner**

*Vedoucí práce:* Ing. David Lindr, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Experimental setup for robotic vision

## Master thesis

*Study programme:* N2612 – Electrical Engineering and Informatics  
*Study branch:* 3906T001 – Mechatronics  
*Author:* **Bc. Jakub Hegner**  
*Supervisor:* Ing. David Lindr, Ph.D.



## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Hegner**  
Osobní číslo: **M14000199**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Mechatronika**  
Název tématu: **Realizace pracoviště robotického vidění**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Seznamte se s problematikou rozpoznávání obrazu a řízení průmyslového robotu ABB.
2. Navrhněte a realizujte systém řízení pásového dopravníku s možnostmi komunikace s nadřazeným systémem.
3. Vytvořte software pro rozpoznávání polohy předmětu čtvercové podstavy na pohybujícím se dopravníkovém pásu pomocí jedné stacionární kamery.
4. Navrhněte trajektorie průmyslového robotu s proměnnými parametry jedné osy kartézského systému. Dále navrhněte způsob předávání informace o zjištěné poloze pohybujícího se předmětu na dopravníkovém pásu, díky které bude možné sbírat robotem předměty z dopravníkového pásu.
5. Vypracujte závěrečnou dokumentaci.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: 40–50 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Craig J., J.: Introduction to Robotics Mechanics and Control, Third Edition, Prentice Hall, 2005.
- [2] Delores M. Etter and Jeanine A. Ingber.: Engineering Problem Solving with C++, 3rd edition, Prentice Hall Press, Upper Saddle River, NJ, USA, 2011.
- [3] Dietrich W. R. Paulus, Joachim Hornegger: Pattern Recognition and Image Processing in C++, 1st edition, Verlag Vieweg, Braunschweig/Wiesbaden, 1995.

Vedoucí diplomové práce:

**Ing. David Lindr, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce:

**doc. Ing. Josef Černohorský, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: 10. října 2016

Termín odevzdání diplomové práce: 15. května 2017

prof. Ing. Zdeněk Pliva, Ph.D.

děkan



doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2016

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

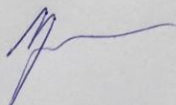
Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15.05.2017

Podpis:





## **Poděkování**

Rád bych poděkoval vedoucímu mé Diplomové práce Ing. Davidu Lindrovi, Ph.D. za rady, věcné připomínky, odborný dohled a především trpělivost. Za totéž bych chtěl poděkovat i doc. Ing. Josefu Černožorskému, Ph.D. Dále bych rád poděkoval Ing. Davidu Krčmaříkovi, Ph.D. za odborné rady spojené s programováním v LabView a též Janu Štruplovi za kontrolu návrhu elektrického zapojení rozvaděče. V poslední řadě bych rád poděkoval rodině za pevné nervy a podporu.

## Abstrakt

Tato Diplomová práce se zabývá řízením dopravníkového pásu, a především návrhem software pro rozpoznávání předmětu na dopravníkovém pásu a následné spolupráce s řídicím systémem průmyslového robota ABB. V úvodu práce jsou rozebrány jednotlivé použité komponenty a problematika rozpoznávání obrazu týkající se kamer, prostředí či samotného software pro rozpoznávání. Dále také řízení průmyslového robota včetně konkrétní specifikace použitých periférií. Následující kapitola se věnuje návrhu a realizaci řízení dopravníkového pásu – kompletní schématické zapojení a program pro PLC řídicího pohyb dopravníku. Zmíněna jsou i omezení, která limitovala podobu dopravníku. Tato část též uvádí jak návrh a zprovoznění řídicího software pro rozpoznávání předmětu na zhotoveném dopravníkovém pásu, tak i jeho napojení na řídicí systém robota ABB. Závěrem je zhodnocena funkčnost a účinnost celého systému, jsou zmíněny problémy, které bylo třeba řešit a je zdůvodněno, proč bylo použito právě toto řešení. Též jsou konkretizovány podmínky, při kterých je úloha nejefektivnější a kdy naopak efektivita klesá. Nechybí ani návrhy na zlepšení, případně i alternativní možnosti řešení ať už konkrétních částí nebo celku.

**Klíčová slova:** software, dopravníkový pás, průmyslový robot, rozpoznání obrazu.

## **Abstract**

This master's thesis is focused on conveyor belt control and mainly on software design for object recognition on it. Talks about cooperation with control system of industrial robot ABB. At the beginning of the thesis are analyzed used components and problems of image recognition related to cameras, environment or recognition software itself. Furthermore, the industrial robot control also includes a specification of the used peripherals. Next part is focused on design and implementation of conveyor belt control. Limitations that limit the shape of the conveyor are also mentioned. This section also introduces the design and commissioning of the control software for object recognition on a conveyor belt, as well as its connection to the ABB robot control system. The following chapter deals with the conveyor belt design and implementation – a complete schematic connection and program for the conveyor control PLC. Limitations that limit the shape of the conveyor are also mentioned. This section also introduces the design and commissioning of the control software for object recognition on a conveyor belt, as well as its connection to the ABB robot control system. Finally, the functionality and effectiveness of the entire system is evaluated, the problems that need to be addressed are mentioned, and the reason for usage of this solution. Concrete conditions are also specified in which the task is most effective and on the contrary the efficiency decreases. Suggestions for improvements as well as alternatives to solving specific parts or whole system.

**Keywords:** software, image recognition, industrial robot, conveyor belt, design.



# OBSAH

<b>1. ÚVOD .....</b>	<b>12</b>
<b>2. POPIS APLIKACE A JEHO ČÁSTÍ.....</b>	<b>13</b>
2.1. USPOŘÁDÁNÍ APLIKACE .....	13
2.2. VOLBA DOPRAVNÍKU .....	14
2.3. MOTOR A PŘEVODOVKA.....	15
2.4. PLC X20CP13XX .....	17
2.5. SENZORY POLOHY OMRON E3Z – T61/T66 .....	18
2.6. KAMERA – BASLER ACE ACA1300-30GC .....	18
2.7. CCD SENZORY .....	19
2.7.1. Princip fungování .....	20
2.7.2. Organizace pixelů v CCD čipu.....	20
2.7.3. Binning .....	21
2.7.4. CCD videokamery.....	22
2.8. SENZOR KAMERY .....	22
2.8.1. Vysoká citlivost a nízká rozmazanost .....	23
2.8.2. Rozmanitá možnost provozu .....	23
2.9. GIGE VISION .....	23
2.9.1. Současná verze 2.0 .....	24
2.10. ROZPOZNÁNÍ OBRAZU NA POHYBLIVÉ SCÉNĚ.....	25
2.11. LABVIEW .....	26
2.11.1. O LabView .....	26
2.11.2. Využití .....	26
2.11.3. Přednosti.....	27
2.12. TCP/IP PROTOKOL .....	27
2.12.1. Aplikační vrstva (Application Layer) .....	27
2.12.2. Transportní vrstva (Transport Layer) .....	28
2.12.3. Síťová vrstva (Internet Layer) .....	28
2.12.4. Vrstva síťového rozhraní (Network Interface Layer).....	28
2.13. SOCKET FUNKCE .....	29
2.14. ŘÍDICÍ SYSTÉM ROBOTA (ROBOT CONTROLLER).....	29
2.14.1. S4C .....	30
2.14.2. S4C plus.....	30
2.14.3. IRC5.....	31
2.15. ROBOTY .....	32
2.15.1. IRB 140.....	32

2.15.2. IRB 1400.....	32
2.15.3. IRB 120.....	33
2.16. NÁSTROJE .....	34
2.16.1. Chapadlo.....	34
2.16.2. Přísavka .....	34
2.17. PROGRAMOVÉ PROSTŘEDÍ – ROBOTSTUDIO.....	35
2.18. MOŽNOSTI KOMUNIKACE S ŘÍDÍCÍM SYSTÉMEM ROBOTA.....	36
<b>3. PROGRAMY A JEJICH VZÁJEMNÉ PROVÁZÁNÍ.....</b>	<b>37</b>
3.1. ELEKTRICKÉ ZAPOJENÍ ROZVADĚČE DOPRAVNÍKU.....	37
3.2. PLC PROGRAM.....	37
3.3. PROGRAM V LABVIEW .....	39
3.3.1. Ovládací rozhraní.....	45
3.4. PROGRAM ROBOTA .....	46
3.5. VYTVOŘENÍ A NAHRÁNÍ SYSTÉMU ROBOTA .....	49
3.6. ODEBÍRÁNÍ PŘEDMĚTU .....	51
<b>4. PROBLÉMY .....</b>	<b>52</b>
4.1. PŘEVODOVKA .....	52
4.2. ZPOŽDĚNÉ ZPRACOVÁNÍ HRAN.....	52
4.3. VYTVOŘENÍ SYSTÉMU ROBOTA.....	52
4.4. NAHRÁNÍ SYSTÉMU ROBOTA .....	52
4.5. ZAZNAMENÁNÍ OBJEKTU NA DOPRAVNÍKOVÉM PÁSU.....	53
<b>5. ZÁVĚR.....</b>	<b>54</b>
<b>CITACE .....</b>	<b>56</b>
<b>PŘÍLOHY.....</b>	<b>58</b>
ELEKTRICKÉ ZAPOJENÍ ROZVADĚČE A PLC .....	58
OBSAH CD .....	60

## Seznam obrázků

Obrázek č. 1. Celkové uspořádání .....	13
Obrázek č. 2. Pásový dopravník [5] .....	14
Obrázek č. 3. Článekový dopravník [6] .....	15
Obrázek č. 4. Motor s převodvkou SK 63 S/4 [8] .....	16
Obrázek č. 5. PLC [9] .....	17
Obrázek č. 6. Senzor polohy [10] .....	18
Obrázek č. 7. Kamera Basler Ace [12] .....	18
Obrázek č. 8. GigE Vision Packet [16] .....	25
Obrázek č. 9. Řídicí systém robota S4C [19] .....	30
Obrázek č. 10. Řídicí systém robota IRC5 [21] .....	31
Obrázek č. 11. ABB IRB 140 [23] .....	32
Obrázek č. 12. ABB IRB 1400 [24] .....	33
Obrázek č. 13. ABB IRB 120 [25] .....	33
Obrázek č. 14. Chapadlo Festo .....	34
Obrázek č. 15. Přísavka .....	35
Obrázek č. 16. Diagram programu PLC pro jednosměrný pohyb .....	38
Obrázek č. 17. Funkce Vision Acquisition a její nastavení .....	41
Obrázek č. 18. Vyhodnocení min a max hodnot souřadnic v LabView .....	42
Obrázek č. 19. Výřez snímku z kamery po spočítání min a max na ose X a Y .....	43
Obrázek č. 20. Souřadnice pro přepočet .....	45
Obrázek č. 21. Ovládací rozhraní v LabView .....	46
Obrázek č. 22. Vytvoření virtuální jednotky .....	47
Obrázek č. 23. Vytvořený virtuální signál .....	47
Obrázek č. 24. Model programu robota .....	48
Obrázek č. 25. Komunikace na straně robota .....	48

Obrázek č. 26. Přetypování a ověření příchozí hodnoty .....	49
Obrázek č. 27. Přiřazení poslané hodnoty robtargetu.....	49
Obrázek č. 28. System builder.....	50
Obrázek č. 29. Možnost výběru Options .....	50
Obrázek č. 30. Elektrické zapojení 230V .....	58
Obrázek č. 31. Elektrické zapojení 24V .....	59

# 1. Úvod

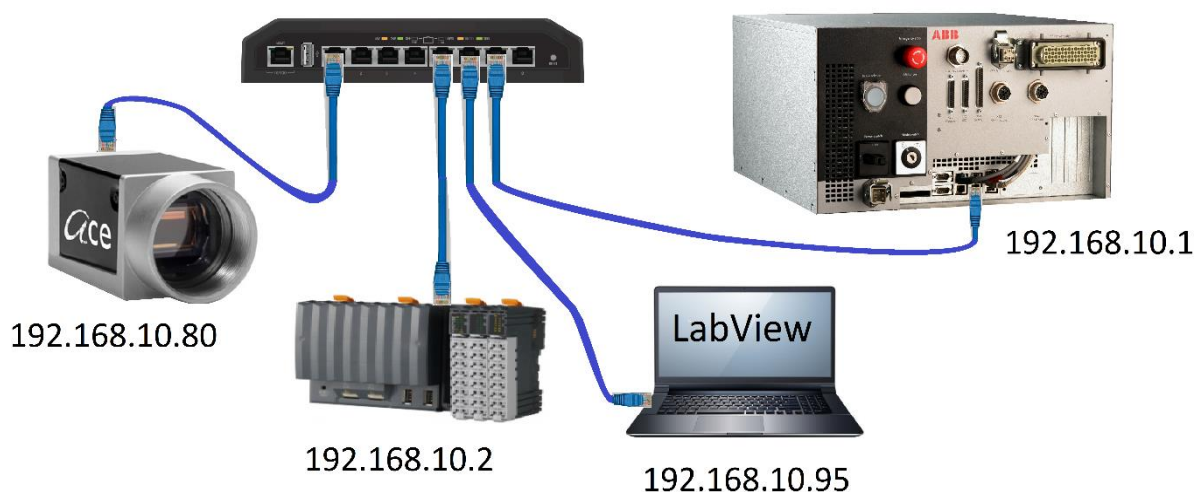
Velké množství firem zabývajících se výrobou či montáží se v současné době snaží urychlit a zoptimalizovat pracovní proces a též snížit cenové náklady. Všechny tyto aspekty dohromady, lze ovlivnit především robotizací a automatizací výrobního postupu, které skýtá velkou řadu možností, počínaje jednoduchým používáním senzorů, konče využitím inteligentní robotiky jako takové. Velice zajímavou variantou je využití senzorů, dopravníku a průmyslového robota s využitím kamer. Pro vyhodnocení a celkovou funkčnost zde může fungovat a zpravidla také funguje PLC, které pomocí jednoduchého programu zvládne vyhodnocovat informace ze senzorů a robota, na jejichž základě řídí měnič, a tedy i celý dopravník. Inteligentní kamera je schopna pracovat s průmyslovým robotem jako samostatná jednotka, která robotu udává přesné informace o hledaném objektu nacházejícím se na dopravníku a vše funguje jako komplexní celek.

Tomuto propojení se věnuje diplomová práce: Realizace pracoviště robotického vidění. Zvolené téma je zajímavé především s ohledem na využívání podobného modelu v průmyslu a též pro vzájemné propojení několika oborů, které zahrnuje. Ať už se jedná o propojení počítačového vidění s robotikou, kde inteligentní kamera předává přesné souřadnice a natočení pohybujícího se objektu průmyslovému robotu a tím mu umožní tento objekt přesně zachytit ve stanovené poloze a následně s ním pracovat. Či propojení automatizace, robotizace a elektroniky, při kterém PLC na základě potřeb průmyslového robota a informací ze senzorů posílá do měniče potřebné informace pro ovládání pohybu motoru pásového dopravníku a udávání jeho rychlosti. Nicméně je potřeba na tuto úlohu nahlížet jako na uzavřený systém, kde změna u jedné části může, a obvykle ovlivní část jinou, a právě toto vzájemné propojení se jeví velice zajímavě. Práci se jednak zabývá návrhem a sestavením tohoto systému jako funkčního celku, tak i problematikou a omezeními, které je potřeba řešit u jednotlivých částí či periférií. Cílem je tedy vytvoření funkčního robotického pracoviště, využívajícího několik technických oblastí současně.

## 2. Popis aplikace a jeho částí

### 2.1. Uspořádání aplikace

Celá aplikace je tvořena ze čtyř základních částí, které spolu vzájemně komunikují. Jedná se o kameru Basler Ace acA1300–30gc, průmyslového robota ABB IRB 120 který je ovládán řídicím systémem IRC5, PLC B&R X20CP1381 a počítač, který je využit pro zpracování obrazu – v tomto případě pomocí prostředí LabView. PLC díky analogové výstupní kartě posílá informace do měniče B&R Acopos Inverter s44 a získává informace od senzorů polohy Omron E3Z pomocí digitálních signálů, na něž reaguje a v případě potřeby je posílá dál po TCP/IP komunikaci do LabView, eventuálně samo nastavuje výstupní signály/hodnoty. LabView zajišťuje udržení komunikace mezi ním a PLC a na základě informací z něj aktivuje kameru a získává z ní snímky. Ty následně vyhodnotí a pošle robotu informaci o objektu, který je třeba odebrat z jedoucího dopravníku. Robot též komunikuje s LabView pomocí TCP/IP. Dále bylo nezbytné, aby byly všechny komponenty ve stejné podsíti. Celý přehled probíhajících komunikací a jednotlivé IP adresy naleznete na Obrázek č. 1 Celkové uspořádání. Podrobnějšímu rozboru jednotlivých komponent se bude věnovat celá tato kapitola. Popis funkcí jednotlivých zařízení, jejich programů, nastavení a problémům s tímto spojeným naleznete v kapitolách následujících.



Obrázek č. 1. Celkové uspořádání

## 2.2. Volba dopravníku

V průmyslu existuje celá řada dopravníků, které lze rozdělit dle mnoha parametrů, především pak podle umístění a typu práce, kterou mají vykonávat. Naším cílem bylo zajistit nepřetržitou dopravu kusových objektů po předem stanovené dráze tak, aby je bylo možné odebírat za chodu přilehlým průmyslovým robotem a pokud možno jej zakomponovat do laboratoře robotiky bez omezení práce na ostatních aplikacích v laboratoři. Pro toto využití přicházely v úvahu vesměs dva typy dopravníků.

- Pásové: Jsou určeny k plynulé dopravě sypkých, zrnitých, nelepivých materiálů, popřípadě kusového zboží – pytlů, palet, kartonů apod. Doprava může být ve vodorovném, nebo šikmém směru. Dopravní vzdálenost je od několika metrů až po stovky metrů. Nosná konstrukce je zpravidla tvořena z oceli. Hnací ústrojí je poháněno elektromotorem s převodovkou. Pás, který plní nosnou i tažnou funkci, je natahován napínacím bubnem se závažím a unášen třením, vzniklým ve styčných plochách pásu a hnacího bubnu. Jeho materiálem je nejčastěji pryž s textilní vložkou, používají se též plasty či ocel a jeho správnou polohu v příčném směru zajišťují válečky. [4]

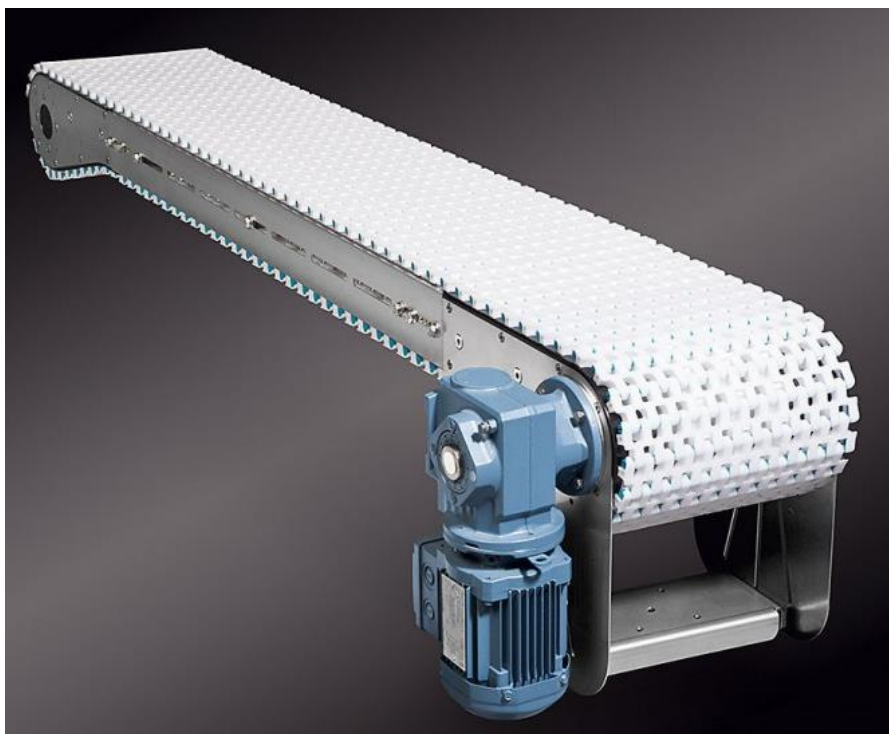


*Obrázek č. 2. Pásový dopravník [5]*

- Článekové: Používají se pro přepravu těžších břemen při menších rychlostech než u pásových dopravníků, především kusových předmětů o větší hmotnosti, hrubých, ostrohranných i horkých materiálů. Též je lze řadit do technologických procesů. Hlavní část tvoří dva řetězy, které mají na čepech uložené kladky, valící se po vodítkách –



pevné dráze. Tyto řetězy zabírají do hnacích a hnaných řetězových kol, z nichž hnaná kola jsou spojena s napínacím zařízením. Pohon jde od elektromotoru přes převodovku na hřídel hnacích kol, k jednotlivým článkům řetězů jsou přivedeny nosné dřevěné nebo ocelové desky k dopravě břemen. Úhel sklonu těchto dopravníků může být až 45°. [4]



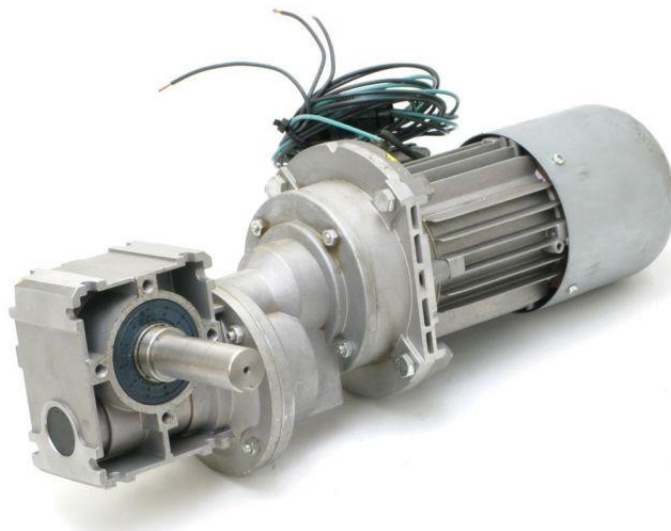
*Obrázek č. 3. Článekový dopravník [6]*

Po zvážení možností bylo poptáno několik firem na vyhotovení kalkulace a zvolila se varianta dopravníku pásového, především s ohledem na jeho cenu, která činila desetinu ceny článkového dopravníku.

Vzhledem k tomuto rozhodnutí, bylo potřeba zajistit trvalý chod bez přepadávání předmětů přes okraj jiným způsobem než v případě oválné dráhy článkového dopravníku, kde by se tento problém řešit nemusel.

## 2.3. Motor a převodovka

Pro pohánění dopravníkového pásu je použit tří fázový, čtyř stupňový, asynchronní motor s čelní tří stupňovou převodovkou NORD SK 63S/4 [7]



*Obrázek č. 4. Motor s převodkou SK 63 S/4 [8]*

Parametry motoru:

- Krytí: IP 55
- Napájecí frekvence: 50Hz
- Napájecí napětí: 230V
- Rychlost: 1335ot/min
- Výkon: 0,12kW
- Fázový posun  $\cos\phi$ : 0,64

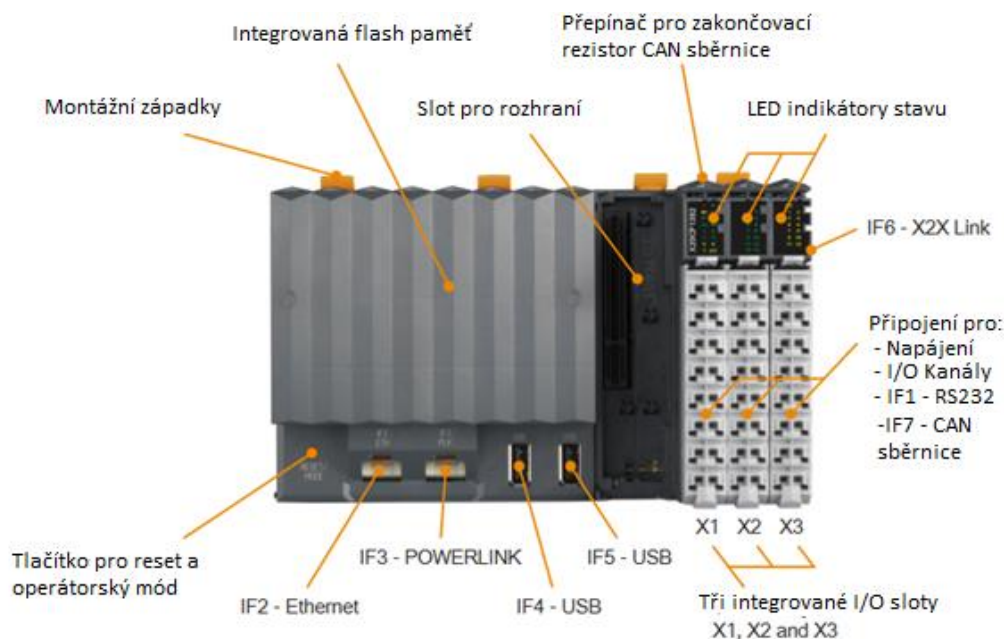
Parametry převodovky:

- Moment na výstupní hřídeli: 69Nm
- Hnací výkon: 0,12kW
- Převodový poměr: 1000
- Provozní faktor: 0,8

## 2.4. PLC X20CP13XX

Řada kompaktních procesorů X20 obsahuje tři integrované vstupně – výstupní moduly. Přístroj je vybaven třiceti různými digitálními vstupy/výstupy a dvěma analogovými vstupy, přičemž jeden analogový vstup lze využít pro měření teploty pomocí PT1000. [9]

Procesor ve verzi X20CP1301, který je využit pro tuto aplikaci, je rovněž vybaven jedním rozhraním Ethernet a též jedním USB. Ve vyšších verzích jsou k dispozici i integrovaná rozhraní POWERLINK a CAN-Bus. Jsou-li zapotřebí další sběrníkové spoje, procesor může být rozšířen modulem ze standardního sortimentu X20. Této možnosti je zde využito a procesor je tak rozšířen o výstupní analogovou kartu X20AO4622, která je nutná pro posílání hodnot do měniče. X20CP1301, který je použit v této aplikaci, je vybaven kompatibilním procesorem Intel x86 200MHz, 128MB RAM a vestavěným flash diskem o kapacitě 1GB. FRAM pro ukládání remanentní proměnné, má k dispozici 16kB paměti. Návrh, jenž neobsahuje ventilátor ani baterii zaručuje bezúdržbový provoz. Slot pro rozhraní, jenž je k dispozici není v současné době nijak využíván a je zaslepen krytkou.



Obrázek č. 5. PLC [9]

## 2.5. Senzory polohy Omron E3Z – T61/T66



*Obrázek č. 6. Senzor polohy [10]*

Omron E3Z se vyrábí v několika variantách. V případě T61/T66 se jedná o jednocestný model, který dokáže snímat do vzdálenosti deseti metrů. Tento senzor též zaručuje vysokou stabilitu díky unikátnímu algoritmu, který chrání proti rušení vlivem vnějšího světla. [11]

## 2.6. Kamera – Basler ace acA1300-30gc



*Obrázek č. 7. Kamera Basler Ace [12]*

Společnost Basler pokrývá celou škálu kamer od levných, po velmi rychlé kamery s vysokým výkonem při malých rozměrech 42 x 29 x 29 mm. Série Ace využívá CCD a CMOS snímače s rozlišením až 5 megapixelů a propustností 340 snímků za sekundu, a to jak černobíle, tak i barevně. Basler ace acA1300–30gc je určen především nízkou cenou, i tak se snaží udržet standardní kvalitu a využívá znalosti předchozích sérií.

Kamera z řady Basler Ace je určena pro průmysl, zdravotnictví a dopravní aplikace, kde jsou požadavkem co nejmenší rozměry a velmi nízká cena, výhodou jsou též provozní teploty a propustnost snímků za sekundu. Kamery jsou vhodné také pro použití přímo na ramenu robota.

V sérii Ace je možné volit ze dvou datových rozhraní, a to CameraLink a GigE Vision, který je založen na standardu Ethernet. Kameru komunikující přes rozhraní GigE Vision je možno napájet buď přes 6-ti pinový konektor Hirose pomocí externího zdroje 12VDC, nebo přes ethernet pomocí technologie Power Over Ethernet (PoE). K tomu je potřeba použít buď síťovou kartu, nebo jiný síťový prvek, který tuto technologii podporuje. Lze také využít adaptér PoE SinglePort Injector, který převede „obyčejný“ ethernet na PoE variantu. Díky gigabitovému ethernetu je možné kameru použít v klasické síťové infrastruktuře i na větší vzdálenosti. S použitím aktivních síťových prvků je možné k jednomu počítači připojit více kamer najednou.

Použitý čip Sony ICX409 pracující v prokládaném režimu je hojně používán v analogových kamerách. Výměna analogové kamery za digitální tak přináší veškeré výhody digitálních kamer s minimem komplikací při jejich fyzické výměně i vzhledem k jejím rozměrům. Součástí kamery je standardně závit pro objektiv typu C, volitelně lze objednat CS.

Kamera má několik nastavitelných parametrů a zajímavých vlastností jako jsou: zesílení A/D převodníku, nastavení vyvážení bílé, volně nastavitelná oblast zájmu (AOI), nastavitelná doba expozice a snímací frekvence, gama korekce, binning, bitová hloubka 8, 10 nebo 12 bitů, automatické zesílení a doba expozice, interní nebo externí synchronizace, možnost napájení po Power Over Ethernet. [13]

## 2.7. CCD Senzory

CCD (Charge – Coupled Device) je nejčastěji používaný, velmi citlivý obrazový senzor složený z polovodičových buněk, které mají lineární odezvu na osvětlení. Výhodou je především jejich mechanická stabilita a poskytování snímků v digitální podobě, tudíž jsou velmi vhodné pro využití k následnému zpracování. Jeho výroba je sice jednoduchá, ale poměrně nákladná.

### 2.7.1. Princip fungování

Nábojově vázané prvky sice vznikly jako experimentální počítačové paměti, ale jejich schopnost převádět světlo na elektrický signál z nich udělala nejlepší známé detektory světla. Základní princip práce CCD je poměrně jednoduchý – přicházející světlo vytváří v polovodiči elektrický náboj (elektrony). Elektrony se nemohou volně pohybovat po čipu, neboť na čipu jsou vytvořeny svíslé negativní potenciálové valy (odpuzející elektrony). Systém vodorovných elektrod, rovněž s negativním nábojem, vytváří na čipu mřížku tzv. „potenciálových studní“, z nich elektrony nemohou uniknout. Každá potenciálová studna vytváří reprezentuje jeden obrazový bod (tzv. „pixel“ z anglického picture element), tedy nejmenší čtvereček obrazu. Počet pixelů v horizontálním a vertikálním směru stejně jako velikost pixelu tvoří jedny z nejzákladnějších charakteristik CCD čipu. Pixely vystavené většímu množství světla naakumulují více elektronů a naopak. Jedna ze základních výhod CCD čipů ve srovnání s lidským okem tedy je schopnost akumulace náboje po dlouhou dobu. CCD tak mohou postupně nashromáždit dostatek světla i z velmi slabých světelných zdrojů. [14]

Jak již bylo řečeno, CCD čip je pokryt sítí elektrod, která udržuje světlem uvolněné elektrony v pixelech. Ale struktura elektrod je poněkud komplikovanější. Pokud se na elektrody přivede různé napětí, elektrony mohou být „přelévány“ z jedné nábojové studny do sousední. Tak je možné náboj posouvat po ploše čipu. Tento proces je používán, když je potřeba informaci z CCD čipu vyčíst. Balíky elektronů, reprezentující jednotlivé pixely, jsou posouvány do výstupního zesilovače, kde je elektrický náboj převeden na napětí. Toto napětí se objeví na výstupním pinu CCD čipu. Elektronika kamery pak musí toto napětí změřit (převést na číslo pomocí analogově/digitálního převodníku, anglicky Analog/Digital Converter – ADC) pro každý pixel. Informace o náboji akumulovaném v každém pixelu (a tedy o množství světla, která do každého pixelu dopadlo) tvoří datový soubor reprezentující obrázek. [14]

### 2.7.2. Organizace pixelů v CCD čipu

Jediná řada pixelů tvoří lineární CCD. Lineární CCD bývají používány, když je zapotřebí snímat pouze jednorozměrný obraz (např. při detekci spektra). I dvourozměrný obraz ale může být vytvořen s pomocí lineárního snímače – stačí aby se detektor nebo cílový objekt pohyboval a obraz je vytvořen postupným snímáním řádek po řádku. Tento princip je použit např. u skeneru dokumentů – detektor spolu s optikou se posouvá vzhledem ke snímanému dokumentu a obraz se vytváří řádek po řádku. Také kamery v družicích a sondách obíhajících

kolem zkoumaných planet často používají tento princip – orbitální pohyb družice je použit k vytváření obrazu povrchu řádek po řádku. [14]

Pixely vytvořené v matici na křemíkovém čipu vytvářejí maticový CCD. Maticový CCD detektor snímá obrázek najednou. Používají se v digitálních fotoaparátech, kamerách a také v astronomických CCD kamerách. [14]

Pixely akumulující světlo jsou v maticových CCD organizovány do sloupců. Změnami napětí na vertikálních elektrodách je možné celý obraz posunout podél obrazových sloupců o jeden řádek dolů. To znamená, že každý řádek se posune o jeden řádek dolů, pouze nejnižší řádek se přesune to tzv. horizontálního registru. Horizontální registr může být posouván pomocí horizontálních elektrod do výstupního zesilovače. Čtení maticového CCD tedy představuje svislý posun obrazu do horizontálního registru prokládaný vodorovným posunem pixelů horizontálního registru do výstupního zesilovače a digitalizací jeho hodnoty. [14]

### 2.7.3. Binning

Jedna z důležitých vlastností CCD čipů, často využívaná v astronomických aplikacích, je schopnost „slévat“ náboje okolních pixelů během vyčítání (anglicky je toto slévání označováno binning). Náboj digitalizovaný ve výstupním uzlu tak představuje součet osvětlení pixelů slitých dohromady. [14]

Pixely je možné slévat v horizontálním i vertikálním směru. Horizontální binning vznikne, když je horizontální registr posunut vícekrát, aniž by byl mezi jednotlivými posuny inicializován výstupní bod čipu. Vertikální binning vznikne, když je obraz vertikálně posunut do horizontálního registru, aniž by byl mezi posuny horizontální registr vyčten. Kombinace horizontálního i vertikálního binningu vede k nahrazení čtverce nebo obdélníku sousedních pixelů jediným bodem obrazu. Například binning  $2 \times 2$  je kombinací  $2 \times$  vertikálního a  $2 \times$  horizontálního binningu. Maximální počet pixelů, které je možno slít, zpravidla závisí na elektronice kamery a na jejím firmware. Některé kamery dovolují jen určité předdefinované kombinace binningu (např.  $2 \times 2$  a  $3 \times 3$ ), případně dovolují libovolně kombinovat binning v určitém rozsahu (např. 1–4 v horizontálním a 116 ve vertikálním směru apod.).

Výhoda vyčítání CCD čipu s binningem je tedy následující. Rozlišení výsledného obrazu (počet pixelů) je menší, ale citlivost čipů výrazně roste. V případě binningu  $2 \times 2$  představuje každý bod obrazu součet 4 pixelů a signál je tedy  $4 \times$  větší. Binning je také velmi užitečný, pokud jsou pixely kamery podstatně menší než nejmenší detail, který daný dalekohled



dokáže zobrazit, ať již z důvodu neklidu vzduchu, dlouhé ohniskové vzdálenosti apod. Binning v takovém případě zvýší citlivost, zrychlí digitalizaci a zmenší objem dat, aniž by omezil úhlové rozlišení – obraz v plném rozlišení by byl zbytečně převzorkovaný. Pro moderní CCD kamery s miliony relativně malých pixelů je binning stále důležitější. [14]

#### 2.7.4. CCD videokamery

Videokamery CCD jsou nejdůležitějšími součástmi video systému, neboť poskytují požadované elektronické obrazové informace, které mohou být zaznamenány na monitoru nebo na videopásku vždy jen v takové kvalitě, jak je zachytí kamera. V dnešní kamerové technice se používají téměř výlučně polovodičové senzory CCD, odolné proti opotřebení, které poskytují nezkreslený obraz bez obtahových efektů. Barevné kamery poskytují více informací. Při pátrání po lidech je např. důležité znát údaje o barvě oblečení a vlasech. Také věk osob lze pomocí barevného obrazu lépe odhadnout. V technické oblasti je často důležité vědět, jakou barvu má materiál nebo tekutina. Rovněž v medicíně jsou nezbytné informace o barvách. Pro rozeznání jemných barevných odstínů je vždy nutné dostatečné osvětlení a vhodná teplota barvy. [15]

### 2.8. Senzor kamery

V současné době vzrůstá poptávka po vysokém rozlišení průmyslových kamer především s ohledem na zabezpečení a dohled nad střeženými objekty. Vyšší rozlišení by pak mělo sloužit pro lepší identifikaci osob zachycených těmito kamerami. Zároveň též existuje trend směřující k převedení konvenčního televizního signálu do digitální podoby. V reakci na tyto potřeby, vyvinula společnost Sony první megapixelový CCD senzor pro bezpečnostní kamery – ICX445AKA.

V tomto zařízení Sony dosáhla jak vysokého rozlišení (1,25 efektivních pixelů) tak vysoké citlivosti. V nové Exview HAD CCD struktuře dosáhly ještě větší citlivosti využitelné pro CCD technologii, a to díky rozsáhlým zkušenostem v oblasti senzorů digitálního fotoaparátu. Tento nový produkt obsahuje také "horizontální a vertikální sčítací funkci pixelů", která umožňuje zachytit rozlišení 60 snímku za sekundu.

ICX445AKA je díky „progressive scan“ s úhlopříčkou 6,0 mm (typ 1/3) a velmi dobrou efektivní barvou pixelů CCD určen pro aplikace bezpečnostních kamer. ICX445AKA nabízí vyšší rozlišení než konvenční CCD snímače bezpečnostních kamer. Kromě toho podporuje výstup signálu v široké řadě formátů.

### 2.8.1. Vysoká citlivost a nízká rozmazanost

Kromě optimalizace pixelové struktury, mikro–čočky na čipu a přijetí doplňkových barevných filtrů, Sony také využila novou generaci Exview HAD CCD technologie, která dosahuje ještě větší citlivosti. V důsledku toho, se podařilo dosáhnout vysoké citlivosti (460mV) i přes malou velikost jednotlivých pixelů (3,75μm). Navíc, se tento senzor může pochlubit vysokou citlivostí blízko infračervené oblasti. Je vhodný pro využití i v noci ve spojení s infračerveným zdrojem světla.

### 2.8.2. Rozmanitá možnost provozu

Obvody ICX445AKA podporují režim vysokorychlostních operací a mohou být provozovány s řídicí frekvencí až do 40,5MHz. To umožňuje tomuto senzoru dosáhnout vysokého rozlišení a velké snímkovací frekvence. V progresivním režimu snímání mohou mít na výstupu všechny pixely nezávislou snímkovací frekvenci až 22,5frame/s. Kromě toho lze výstupní obraz přepočítat na ekvivalentní VGA rozlišení (640 × 480pixelů) s čtyřnásobkem obvyklé citlivosti pomocí funkce horizontálního a vertikálního sčítání pixelů. Sony plánuje další rozvoj této technologie, a to jak v produktech, které využívají především barevných filtrů, tak i ve výrobcích, které podporují černo-bílé zobrazení.

## 2.9. GigE Vision

GigE Vision® je standardní globální rozhraní pro připojení fotoaparátu vyvinuté pomocí Gigabit Ethernet protokolu. GigE Vision umožňuje rychlý přenos snímků pomocí nízkonákladových standardních kabelů na velké vzdálenosti. Díky GigE Vision je bez problémů možné propojit hardware a software od různých výrobců. GigE Vision je rozhraní, které je široce využíváno po celém světě. Desítky předních společností v současné době nabízí stovky výrobků, které odpovídají tomuto standardu.

GigE Vision zahrnuje mnoho výhod:

- FAST – vysoko-pásmový přenos (125MB/s) rychlý převody velkých obrázků v reálném čase  
ADUNDANT – Nekompromisní přenos dat až do délky 100 metrů
- STANDARD – Nízkonákladové CAT5e nebo CAT6 kabely a standardní konektory

- SCALABLE – snadno rozšířitelný pro rychlý Ethernetu
- LOW COST – standardní hardware a kabely umožňující jednoduchost a nízké náklady na integraci

### 2.9.1. Současná verze 2.0

GigE Vision byla aktualizována z verze 1.2 na verzi 2.0 v listopadu 2011. Verze 2.0 obsahuje ovládání zařízení bez datových proudů. Vychází vstříc síťové video distribuci aplikací, u nichž lze využít přepínání sítě klient/server. Zařízení, jako jsou GigE Vision povolují automatické rozpoznání počítače v síti. Verze 2.0 přináší podporu pro rychlejší přenos dat přes 10Gb Ethernet. To také umožňuje přenos komprimovaných obrázků (JPEG, JPEG 2000 a H.264), přesné synchronizace systémů s více kamerami a vylepšenou podporu pro multi-tap senzorů.

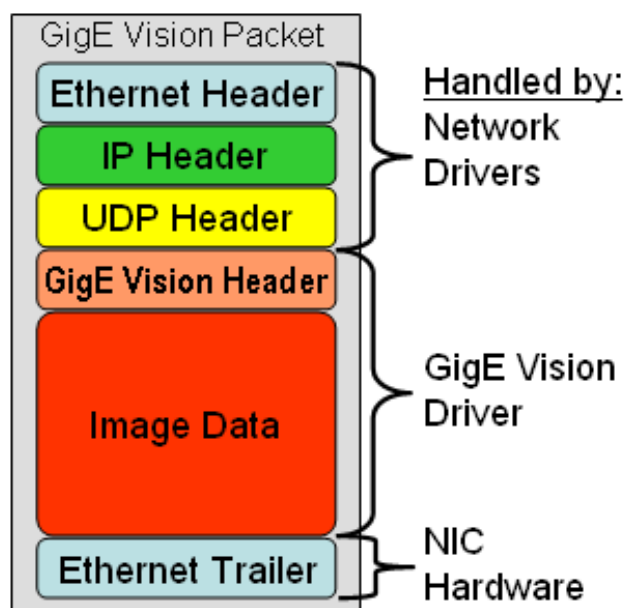
Specifikace GigE Vision se opírá o GenICam™ funkci podporovanou kamerami. Popis této funkce se podobá s XML s ohledem na syntaxi definovanou modulem GenApi specifikace GenICam. GenICam je obecné programovací rozhraní pro všechny druhy kamer.

Aby GigE Vision napomohl dosažení provázanosti mezi různými standardy vidění, zobecnili definici obecných formátů pixelu a vytvořil "Pixel Format Naming Convention". Následně byl tento formát zařazen do standardních funkcí GenICam a pojmenován jako Naming Convection.

Standard GigE Vision byl vyvinut a je udržován specializovanou skupinou společností, které představují jednotlivé části průmyslu zabývající se viděním. AIA dohlíží na vývoj a správu standardu. Aby byla zajištěna plug and play kompatibilita, nesou podporované výrobky logo GigE Vision.

V konečném důsledku má GigE Vision nejlepší účinnost spolu s High performance režimem, který podporují karty Intel PRO 1000 a též umožňuje pracovat i se síťovými prvky jako Switch.

GigE Vision paket pak vypadá následovně:



Obrázek č. 8. GigE Vision Packet [16]

## 2.10. Rozpoznání obrazu na pohyblivé scéně

V dnešní době je veliké množství firem zabývajících se rozpoznáním obrazu a též velké množství programů, ve kterých je možno vytvořit software pro rozpoznání či vyhodnocení obrazu. Tyto možnosti lze rozdělit do několika kategorií.

- Uživatelsky nejprůběžnější je pravděpodobně varianta, kdy je kamera dodána již s hotovým softwarem, který je potřeba pouze upravit adekvátně, k požadovanému vyhodnocení. Takovýto produkty dodává například firma Cognex a pochopitelně si za ně účtuje nemalé částky. Nicméně je jejich řešení velice šikovné, jelikož takováto kamera umí přímo komunikovat s řídicím systémem robota ABB, který na základě vyhodnocení zachytí požadovaný objekt. Další velikou výhodou tohoto řešení je tedy absence jakýchkoli dalších prvků, které by se vyskytovali mezi kamerou a robotem.
- Další možností je rozdělit celou operaci na dvě části. V první části se pouze zachytí obraz, jenž se uloží do obrázku, se kterým může následně operovat jakékoli programovací prostředí, které dokáže s obrázkem pracovat. V lepším případě je možno využít jazyk Python, do něž lze implementovat nespočet vytvořených knihoven pro práci s obrazem a následnou komunikaci. V opačném případě je možné použít i jazyk C, kde je vytvoření softwaru pro vyhodnocení podstatně náročnější.

- Poslední velice zajímavou možností je programové prostředí LabView, které umožňuje jak zachycení obrazu z kamery, tak i jeho následnou úpravu a vyhodnocení. Samozřejmostí je i komunikace s ostatními zařízeními v síti. V LabView je pro práci s obrazem hned několik použitelných funkcí, přičemž jsou velice dobře optimalizované a fungují spolehlivě. Právě z důvodu jednoduchého použití funkcí a také odlišnosti celého prostředí od ostatních programovacích jazyků se LabView jeví jako velice zajímavá volba pro tuto aplikaci.

## 2.11. LabView

### 2.11.1. O LabView

LabView je vývojovým grafickým prostředím, založeným na využití programovacího jazyka G. Toto prostředí je zaměřeno na vývoj aplikací, zajišťujících řízení celého procesu sběru měřených dat, jejich analýzy a prezentace. Jedná se o známý produkt firmy National Instruments, který platí ve světě měřicí a řídicí techniky za standard, s nímž jsou srovnávány ostatní programy. Řada výrobců měřicích a řídicích systémů vyvíjí a dodává ke svým produktům knihovny, které usnadňují použití jejich výrobků, právě při vytváření aplikací v tomto prostředí. [17]

### 2.11.2. Využití

Použití tohoto systému, díky jeho koncepci pro tvorbu měřicích a řídicích aplikací, je výrazné zjednodušení jak při vývoji nové aplikace, tak při jejich následném modifikování. LabView je určeno především pro získávání dat a ovládání přístrojů, proto obsahuje knihovny funkcí a vývojové nástroje navržené speciálně k tomuto účelu. Dnes se dá však použít díky široké paletě dalších knihoven a funkcí i pro obecné programovací úlohy. Aplikace tvořené v LabView jsou nazývány *Virtual Instruments* (VI) (virtuální přístroje), neboť jejich vzhled a činnost připomínají skutečné přístroje. Jejich výkon a použitelnost jsou na rozdíl od tradičních přístrojů omezeny pouze výkonem použitého počítače a k němu připojeného hardwaru. Z výše popsaných vlastností je zřejmé, že moje práce v LabView není standardní pro toto prostředí, můžeme ji tedy považovat za experiment, dá-li se tento programovací jazyk použít pro potřeby geodézie. [17]

### 2.11.3. Přednosti

Výrobce National Instruments tvrdí, že LabView je jediným grafickým programovým prostředím, které obsahuje kompilátor generující optimalizovaný kód, jehož rychlost vykonání je srovnatelná s rychlostí kompilovaných programů v jazyce C. Aplikace vytvořené v LabView jsou plně srovnatelné s aplikacemi vytvořenými i nízko úrovněnými jazyky jako je C, avšak komfort jejich vytváření je podstatně vyšší. Programátor se zbavuje starostí s řadou syntaktických detailů konvenčního programování a může se plně soustředit na řešení zadaného problému. [17]

Další výhodou je rychlost systému, která se projeví například u vytváření inverzních matic. Výrobci počítačových procesorů vyvinuli rychlejší procesory s několika paralelními jádry. LabView dokáže výkonu těchto nových procesorů plně využít ke zvýšení propustnosti testů, pro náročnou analýzu a pro spolehlivý běh aplikací reálného času na vyhrazených jádrech. [17]

## 2.12. TCP/IP protokol

Z ISO/OSI vychází i množina protokolů TCP/IP. Protokol TCP/IP vznikl původně jako komunikační protokol ministerstva obrany USA pro sjednocení počítačové komunikace v rámci ARPANET. Slouží ke komunikaci především v heterogenních sítích. Dnes je součástí prakticky všech operačních systémů (původně byl navrhován hlavně pro UNIX) a je využit ke komunikaci i v síti Internet. Z těchto důvodů vzrůstá jeho význam jako celosvětového standardu. Model TCP/IP je nezávislý na přenosovém mediu a je určen jak pro WAN, tak i pro LAN, jak pro sériové linky, koaxiální kabely, tak i pro vysokorychlostní optické sítě. Je užíván v heterogenní síti (původně určené pro UNIX) Internet. Je to soustava sítí s IP protokolem tvořená mezisít'ovým počítačem. Jednotlivé podsítě mohou být různé (Ethernet, X.25). TCP/IP (Transmission Control Protocol/ Internet Protokol) předpokládá, že na nižších vrstvách jsou pouze nespolehlivé přenosové služby. Zajištění spolehlivosti dělají vyšší vrstvy, a to jen při jejich vyžádání. Rodina protokolů TCP/IP předpokládá existenci čtyř vrstev. [18]

### 2.12.1. Aplikační vrstva (Application Layer)

V této vrstvě jsou provozovány základní aplikace v rámci TCP/IP. Aplikační vrstva zajišťuje přenos a srozumitelnost zpráv. Podle modelu OSI sdružuje vrstvy aplikační, prezentační a relační.

### 2.12.2. Transportní vrstva (Transport Layer)

Obsahuje protokoly TCP a UDP. TCP protokol zajišťuje navázání spojení, zaručení celistvosti zprávy a ukončení spojení. Pokud to požaduje aplikace, tato vrstva pro ni zajistí spolehlivost, tj. změnu nespolehlivý charakter síťové služby na spolehlivý. (komunikuje pomocí portů) V opačném případě (kdy aplikace nevyžaduje spolehlivost od transportní vrstvy nebo vyžaduje rychlost je použit protokol UDP (User Datagram protokol) je jen jednoduchou obálkou nad síťovou vrstvou. UDP nevyžaduje navázání spojení. [18]

### 2.12.3. Síťová vrstva (Internet Layer)

Na této vrstvě pracují protokoly IP, ICMP a na rozhraní se síťovou vrstvou pracují protokoly ARP, RARP. IP protokol se snaží co nejrychleji doručit data (datagram) přes případné mezi uzly až k jejímu koncovému adresátovi, proto nabízí pouze nespolehlivou přenosovou službu, a to nespojovaného charakteru, tzn. IP protokol je nespojovaný nespolehlivý protokol s částečnou detekcí chyb. Tato detekce se týká pouze kontrolního součtu hlavičky protokolu. [18]

IP protokol má tyto funkce:

- Adresování stanic v internetové (síťové) vrstvě
- Definice struktury IP datagramu
- Směrování datagramů
- Propojení internetové a transportní vrstvy. Určitým protokolům jsou přiřazeny pevná čísla portů
- Fragmentace a sestavení datagramů podle MTU – maximum, Transfer Unit – maximální délka datagramu (Ethernet 1500B, seriový port 296B) [18]

### 2.12.4. Vrstva síťového rozhraní (Network Interface Layer)

Zajišťuje přenos rámců (frame) mezi dvěma přímo propojenými počítači. V této vrstvě jsou definované metody přístupu na medium (od koaxiálního kabelu až po optiku). Jelikož zde velmi záleží na konkrétní přenosové technologii (Ethernet, Token Ring, dvoubodový zdroj, telefonní linka apod.), TCP/IP tuto vrstvu nijak blíže nespecifikuje. [18]



V souvislosti s protokoly TCP/IP a jednotlivými vrstvami modelu se rozlišují názvy přenášených jednotek podle toho, jak se váží na jednotlivé vrstvy. [18]

- Rámec – váže se na vrstvu síťového rozhraní a je základní entitou komunikačního media (např. Ethernetu)
- Datagram – internetová vrstva (IP, UDP) – datové kvantum vyměřované mezi stanicemi
- Segment – transportní vrstva (TCP) – proud dat, který je dělen na datagramy.
- Paket – obecně, bez návaznosti na konkrétní vrstvu
- Zpráva – aplikační vrstva [18]

## 2.13. Socket funkce

Socket je jeden konec komunikačního kanálu, přičemž každé aktivní spojení má dva kanály. Každý socket má lokální adresu port a protokol, v případě že je připojen má též vzdálenou adresu a port. Co se týče TCP socketů, které jsou pro nás nejzajímavější, dělí se na dva druhy:

- Naslouchací – pro příjem žádostí o spojení
- Klientský – pro běžnou komunikaci

Algoritmus spojování je pak následující: Jedno zařízení je *host* neboli server, libovolná další zařízení mohou být *client*, přičemž je nezbytné, aby server otevřel naslouchací socket, zadal adresu a port na kterém naslouchá a čekal na spojení. Client pak otevře běžný socket, zadá adresu a port, než se chce připojovat. Tímto připojením obdrží server adresu a port klienta a vznikne mu tím běžný socket. Běžné sockety jsou tímto propojeny a mohou komunikovat. Tato komunikace končí ve chvíli, kdy jedna ze stran ukončí spojení. Vytvořený komunikační kanál je obousměrný a každá strana tak může číst i zapisovat.

## 2.14. Řídicí systém robota (robot controller)

V laboratoři robotiky, pro kterou je navrhována tato aplikace, se vyskytují roboty s následujícími řídicími systémy:

### 2.14.1. S4C

Jedná se o stále používaný funkční systém, který je dostatečně spolehlivý a je možno jej používat mnoha prostředích. Ve školní laboratoři ho používá repasovaný robot IRB1400 M97A. Velkou nevýhodou tohoto systému je disketová jednotka, která je jedinou možností pro export/import dat z/do systému.



Obrázek č. 9. Řídicí systém robota S4C [19]

### 2.14.2. S4C plus

Inovovaná verze systému S4C, která umožňuje komunikaci přes ethernet, případně flash disk, či sériovou linku. Stejně tak jako s prací se systémem S4C je dobré dát si pozor na dlouhé názvy *targetů* či *workobjectů*. V případě, že je program vytvořen v současných verzích programu RobotStudio (5.14, 5.15, 5.60, 5.61) je třeba jej pro správné spuštění ručně upravit. FlexPendant se ovládá pouze pomocí tlačítek a není barevný, tudíž přehlednost, orientace a celková práce s ním nedosahuje tak vysokého standartu jako v případě IRC5. Vzhledově se systém S4C plus od svého předchůdce téměř neliší.

### 2.14.3. IRC5

Tento systém v první řadě umožňuje interaktivní propojení řídicího systému s programem. FlexPendant je dotykový, což velice zjednodušuje a urychluje průběh celého programování, popřípadě úprav programu. Barevné prostředí též napomáhá čitelnosti, přehlednosti a celkové orientace v programu.



*Obrázek č. 10. Řídicí systém robota IRC5 [21]*

Z hlediska urychlení programování, přehlednosti a nespočtu dalších výhod, které poskytuje řídicí systém IRC5 by bylo vhodné mít k dispozici tento systém pro všechny roboty ABB dostupné v laboratoři Robotiky, nicméně i použití starších má své výhody. Ne všechny firmy jsou vybaveny nejmodernější technikou, a tak je třeba, aby byli studenti připraveni i na variantu, že se mohou v praxi setkat se starší technikou. [22] Z hlediska komunikace je vybaven Ethernetovým servisním portem, přes který je možné se k řídicímu systému připojit.

Jak již z popisu jednotlivých řídicích kontrolérů vyplývá, vytvořený software v programu LabView nebude možno využít pro systém S4C, jelikož nepodporuje drátovou komunikaci ať už ethernet nebo sériovou linku, ale bude možné jej využít i pro systém SC4 plus v případě, že se program LabView rozšíří o možnost komunikace po sériové lince. Pro

další dva systémy by mělo být možné využít obě tyto varianty s uživatelskou možností volby. Na rozdíl od standardního řídicího systému robota IRC5 je v laboratoři Technické Univerzity v Liberci použit menší řídicí systém *Compact*.

## 2.15. Roboty

V laboratoři Robotiky na Technické univerzitě v Liberci jsou dostupní čtyři roboti ABB tří různých typů a to IRB 120, IRB 140 a dva roboti IRB 1400. Využitelnost připravovaného programu úzce souvisí s jejich řídicími systémy popsány dříve a také s jejich rozmístěním z hlediska dosahu. Vzhledem k tomu, že se současná kompozice může v budoucnu změnit, nebyl na ni brán ohled a program je za určitých podmínek schopen pracovat s kteroukoli podporovanou jednotkou.

### 2.15.1. IRB 140

Tento průmyslový robot je kompaktním a výkonným šestiosým strojem s unikátní kombinací velkého zrychlení, odolnosti, pracovního dosahu a užitečného zatížení. Vyznačuje se též velkou střední dobou bezporuchového provozu a velmi vysokou opakovatelností. Díky užitečnému zatížení do 6kg a dosahu 810mm je tento průmyslový robot nejlepší ve své třídě.



Obrázek č. 11. ABB IRB 140 [23]

### 2.15.2. IRB 1400

Je známý pro svou tuhost, robustnost, velký pracovní prostor a dlouhý dosah. Je dostatečně rychlý a spolehlivý. Osvědčil se především v obloukovém svařování, manipulacemi

s materiálem či procesními aplikacemi. Dokáže manipulovat s předměty vážícími až 5kg, přičemž je schopen na horní části paže nést doplňkovou zátěž o hmotnosti až 18kg.



*Obrázek č. 12. ABB IRB 1400 [24]*

### 2.15.3. IRB 120

Nejmenší víceúčelový průmyslový robot společnosti ABB váží pouhých 25kg a dokáže manipulovat s břemenem o hmotnosti až 3kg (4kg ve svislé ose) s dosahem ramene 580mm. Jedná se o úspornou a spolehlivou volbu s vysokou výrobní kapacitou a nízkými pořizovacími náklady.



*Obrázek č. 13. ABB IRB 120 [25]*

## 2.16. Nástroje

K odebrání předmětů z jedoucího pásového dopravníku byly využity dva různé způsoby, a tedy dva různé nástroje. Lze tím demonstrovat univerzálnost využití systému.

### 2.16.1. Chapadlo

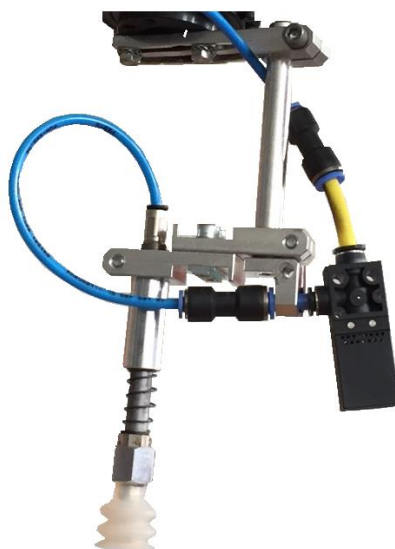
Jedním z nástrojů bylo použito chapadlo od výrobce Festo, které sestává ze tří čelistí, které jsou na ni dodatečně nainstalovány. Čelisti umožňují odebírat předměty kruhové podstavy, případně trojúhelníkové a šestiúhelníkové podstavy. Chapadlo je ovládáno dvěma ventily.



*Obrázek č. 14. Chapadlo Festo*

### 2.16.2. Přísavka

Dalším nástrojem je přísavka, kterých je v laboratoři dostupných hned několik. Všechny, včetně použité fungují na jeden ventil. Výhoda přísavky oproti chapadlu je především v tom, že eliminuje nepřesnosti.



Obrázek č. 15. Přísavka

## 2.17. Programové prostředí – RobotStudio

RobotStudio je program, který umožňuje offline programování, na které je v průmyslu kladen velký důraz, například proto, že při vytváření nebo úpravě programů není potřeba zastavovat provoz reálného robota až do okamžiku, kdy chceme nový software skutečně nahrát.

V RobotStudiosu běží *Rapid Language* paralelně s virtuálním kontrolerem. Tyto programové části spolu úzce spolupracují, ale umí fungovat i nezávisle na sobě. Programy, jež v nich vytváříme, můžeme vytvářet v jedné, či druhé části. Též je možné část vytvářet v *Rapid Language* a část ve virtuálním kontroleru, ovšem je třeba obezřetnosti, aby nedošlo k nechtěnému přepsání jednotlivých částí.

Ke vzájemnému provázání mezi oběma programy nedochází automaticky. V případě, že chceme program vytvořený v RobotStudiosu nahrát do virtuálního kontroleru, používá se funkce *Synchronize to VC*. V případě, že požadujeme opak, použije se funkce *Synchronize to Station*.

RobotStudio dále umožňuje modelovat jednoduchou i složitější geometrii, nicméně je v případě složitější geometrie mnohem lepší vytvořit modely například v programu AutoCad a následně je do RobotStudia implementovat. Dobře využitelnou funkcí RobotStudia je tvorba nástroje.



## 2.18. Možnosti komunikace s řídicím systémem robota

Existuje několik možných způsobů komunikace s robotem. V průmyslových aplikacích se pro komunikaci využívá především Profinet a Profibus, nicméně je možné využít i sériovou linku či TCP/IP.

Vzhledem k četnosti využívání průmyslových komunikací, se Profinet a Profibus staly nedílnou součástí řídicích systémů robota, ale pro jejich využití pro potřeby této práce, by bylo nutné pořídit danou komunikační kartu do počítače. Navíc by tuto kartu muselo podporovat i prostředí LabView. Vzhledem k tomu, že pořizovací cena zmíněných karet není nikterak levná záležitost, bylo nutné komunikaci tímto způsobem zavrhnout.

Jako alternativu lze tedy využít TCP/IP komunikaci, ale pouze za předpokladu, že řídicí systém robota obsahuje, resp. je k němu dokoupen *PC-Interface*, který umožňuje komunikovat mezi robotem a síťovým počítačem. Avšak i toto řešení přináší svá úskalí, jejichž řešení je popsáno v následujících kapitolách. V poslední řadě je možné využít i sériovou linku, od které se však pomalu upouští, na toto krajní řešení však nakonec nedošlo.

## 3. Programy a jejich vzájemné provázání

### 3.1. Elektrické zapojení rozvaděče dopravníku

Do rozvaděče je přivedeno napájení 230V, které je přes 10A jistič přivedeno na měnič Acopos Inverter S44 od firmy B&R. Tento měnič se stará o pohyb asynchronního motoru, který přes převodovku pohání pásový dopravník. Napájení 230V je přivedeno též na zdroj 24V, odkud je toto řídicí napětí rozvedeno do ostatních periférií.

V dokumentaci, kterou naleznete v příloze je též zaneseno připojení PLC včetně příslušných karet. Také je zde připojení jednotlivých vstupů a výstupů karet, jako jsou senzory a přepínače. Nechybí ani signály z analogové karty, které vedou do měniče.

Stavba rozvaděče spolu s pásovým dopravníkem proběhla v rámci mého diplomového projektu, avšak musela nastat jistých změn. Především byly přidány přepínače s aretací. SA1, který umožní samostatně rozjíždět či zastavovat pohyb dopravníku. SA2, jímž se volí režim dopravníku a SA3, jenž umožňuje měnit směr pohybu dopravníku. Tyto jednotlivé piny musí být pochopitelně, stejně jako ostatní vstupy a výstupy, korektně namapovány v programu.

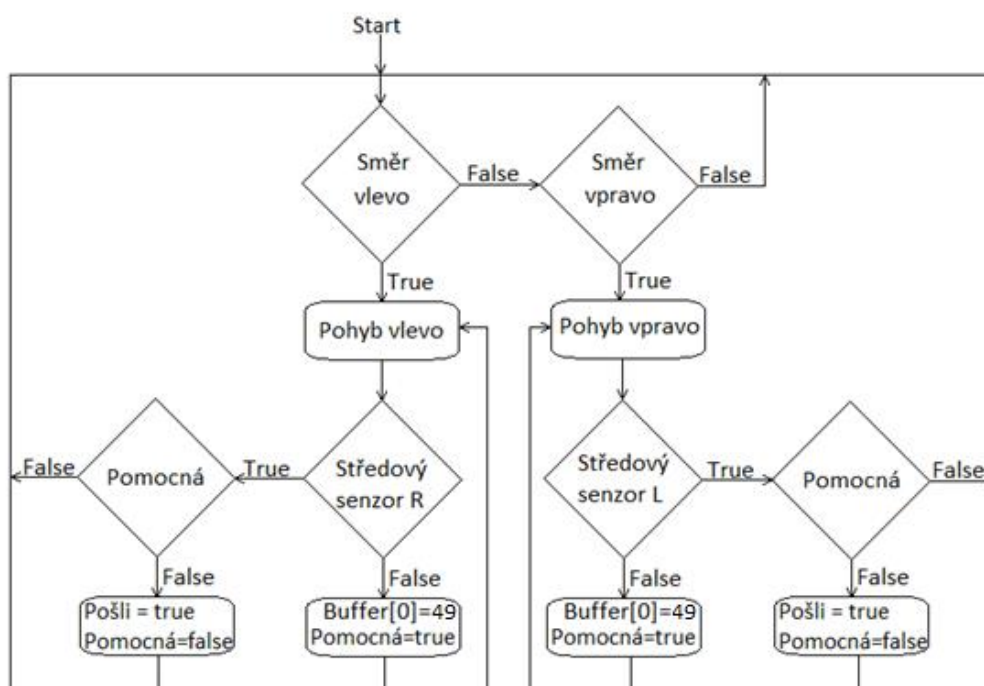
### 3.2. PLC program

Na rozdíl od předchozí verze, která byla vytvořena v rámci mého projektu, kde se pás dopravníku pohyboval pouze jednosměrně, bylo nutné program zdokonalit, k čemuž napomohly zmíněné přepínače pro volby režimů a směrů. Díky této úpravě je nyní možno využívat PLC s přidruženým dopravníkem mnohem efektivněji. Též má dopravník bezesporu lepší využití pro samostatné použití nezávisle na ostatních perifériích, které jsou popsány a používány v této práci. Režimy, které dopravník a PLC umožňují jsou následující:

- Obousměrný – v tomto provozním režimu se pás dopravníku pohybuje ve výchozím stavu vpřed. Pokud je na dopravníkový pás položen objekt, který je dostatečně vysoký, aby jej byly schopny zaznamenat senzory koncové polohy a objekt dosáhne koncové polohy ve směru pohybu dopravníkového pásu, je směr pohybu dopravníkového pásu invertován. V případě, že je senzor koncové polohy aktivován ve směru, při němž nehrozí pád objektu, žádná změna v pohybu dopravníkového pásu nenastane.
- Jednosměrný – v tomto provozním režimu se pás dopravníku pohybuje ve výchozím stavu vpravo za předpokladu, že je přepínač směru ve stavu „rozepnuto“, pokud je

přepínač ve stavu „sepnuto“ je pohyb dopravníkového pásu inverzní, tedy vlevo. V tomto případě není řešena kontrola senzorů koncové polohy – bylo by nutné znát konkrétní požadavky zadavatele aplikace, zda je požadováno zastavení dopravníkového pásu nebo například započtení nezachyceného dílu apod.

V aplikaci, která spolupracuje s kamerou a robotem, je využita pouze varianta jednosměrná, kdy dopravník jede vlevo. Tento koncept je nutný vzhledem k umístění kamery a pásového dopravníku v prostorách laboratoře. Popis funkce jednosměrné varianty programu lze vidět na následujícím obrázku.



Obrázek č. 16. Diagram programu PLC pro jednosměrný pohyb

Také bylo nezbytné přidat a správně vyřešit reakci na jednotlivé senzory. Myšlenka tohoto řešení je patrná z předchozího obrázku, kdy se po přerušení signálu senzoru nastaví do *Bufferu* hodnota „49“, která je určena pro odeslání do LabView, ale samotný signál se odešle až vy chvíli, kdy je signál senzoru opět aktivní (má hodnotu 1). Zde je potřeba zdůraznit, že senzory posílají signál, který je opačný než informace o předmětu, tedy signál je v 0, pokud senzor zaznamená překážku. B&R používá pro komunikaci hodnoty ASCII, z toho důvodu je do proměnné *Buffer[0]* zapsána hodnota 49, nikoli 1. Do LabView přijde hodnota 1.

Dále se do programu přidávala TCP/IP komunikace, která byla z větší části inspirována dostupnou nápovědou. Bylo pouze nezbytné správně pochopit fungování tohoto programu

a vhodným způsobem jej zakomponovat do již existujícího. Výsledek je takový, že program pro komunikaci běží paralelně s programem pro pohyb a vyhodnocení, ale čas jeho cyklu je podstatně kratší. Je nutné si uvědomit a rozmyslet, zda kterou komunikaci chceme použít jako klienta a kterou jako server. Vzhledem k technickým problémům, které se vyskytly při zprovoznování a testování komunikace, bylo nakonec PLC použito jako server, ač by byl vhodnější přesný opak, jelikož v tomto případě je rychlost komunikace a odeslání řídicího signálu závislá na četnosti dotazování v programu LabView a může docházet k drobnému zpoždění, které by mohlo být nežádoucí. Z praktického hlediska je toto zpoždění tak minimální, že se absolutně nemůže negativně projevit na výsledku. Pro správnou funkci bylo potřeba nastavit IP adresu a port pro komunikaci.

### 3.3. Program v LabView

V této aplikaci má program v LabView za úkol zpracovat a vyhodnotit obraz, který poskytuje kamera a zajistit posílání vyhodnocených dat do řídicího systému robota. Celý program funguje na bázi struktur *While*, *Case* a *Queue* – ty se dají chápat jako FIFO registry. Na základě aktuální hodnoty v *Queue* se otevře položka v *Case*, která se vykoná. Zároveň je v každé části *Case* zapisována další hodnota do *Queue* tak, aby byly postupně vykonány všechny potřebné operace. Postupné vykonávání operací v *Case* trvá, dokud je v *Queue* zapsaná hodnota o což se stará funkce *While*.

První část programu se stará o komunikaci s PLC systémem. Jelikož je program v PLC nastaven jako server, je nutné, aby LabView udržovalo spojení s PLC. V opačném případě by v PLC vypršel čas, po který čeká na navázání komunikace. Celá tato komunikační část probíhá nezávisle na zbytku programu – tedy má svou vlastní *Queue*, ze níž se vyčítají hodnoty pouze pro *Case*, ve kterém je zprostředkována komunikace. V případě, že LabView zaznamená signál z PLC, porovná se přijatá hodnota s inicializační hodnotou pro snímání a pokud souhlasí, je snímání skutečně započteno. Programově se tedy zapíše vhodná informace do správné *Queue*. Samotná TCP/IP komunikace je rozdělena na tři části: *PLC init*, *PLC pošli* a *PLC přijmi*.

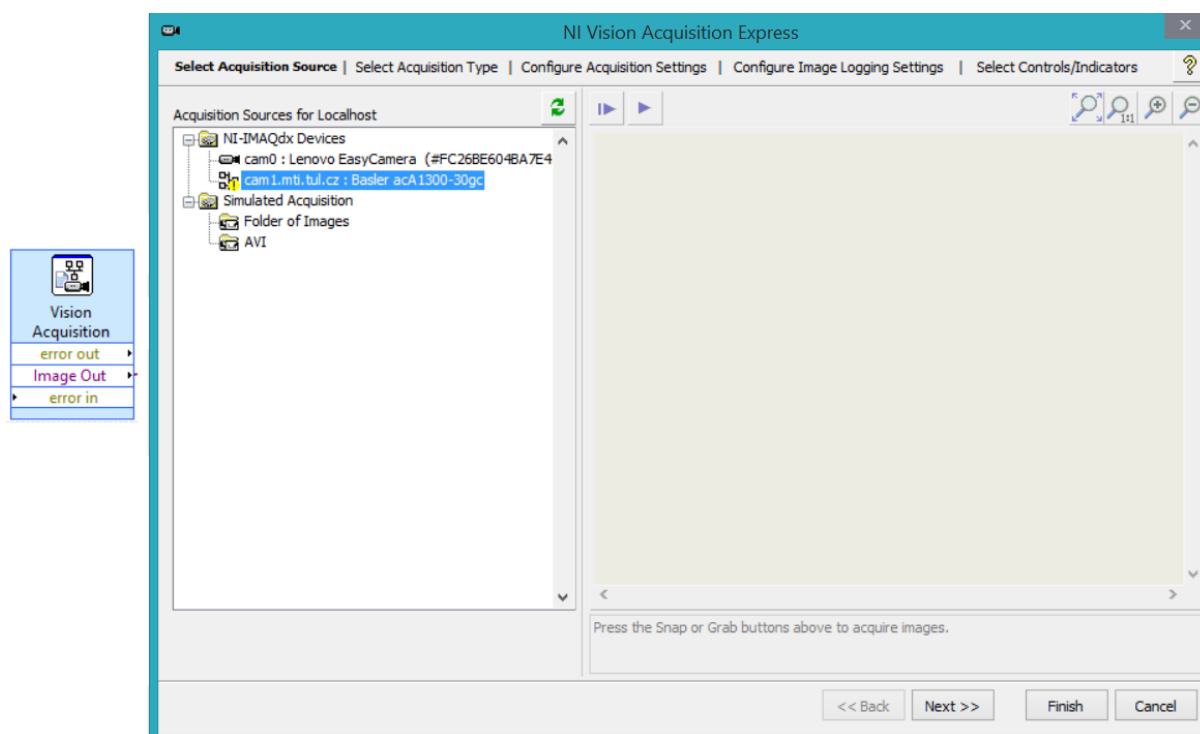
- V části *PLC init* dochází k navázání komunikace s PLC, pro což se využívá funkce *TCP Open Connection*, které je potřeba nastavit především IP adresu a port, na něž se má připojovat. Výstupem z této funkce je pak *Connection ID*, jež si nese informaci o navázaném spojení a je nezbytné pro další pokračování v komunikaci. Jedná se

o jistou identifikaci onoho konkrétního spojení. Do tohoto bloku se program dostane pouze po startu, další nová inicializace není nutná, jelikož je spojení neustále udržováno.

- V části druhé (*PLC pošli*) je použita funkce *TCP Write*, která je nezbytná pro udržení komunikace s PLC. Proto je tato funkce použita v každém cyklu. Pro udržení komunikace stačí odesílat jakoukoli nenulovou hodnotu v podobě *stringu*. Stačí využít například funkce *Number To Decimal String*, jejíž význam je evidentní z jejího pojmenování.
- Poslední část komunikace (*PLC přijmi*) je nejzajímavější, jelikož zde dochází i ke zpracování příchozí informace. Ta je nejprve přečtena funkcí *TCP Read* abymohla být posléze přetypována na číslo. Jelikož se z PLC programu očekává hodnota „1“, která je ve *stringu*, porovnává se tato přetypovaná hodnota s jedničkou. V případě, že skutečně rovnost nastane je do *Queue*, která se stará o vyhodnocení zapsána textová hodnota *Snímání*, čímž se dostáváme k vyhodnocovací části.

Pro navázání spojení s kamerou se hlavní otěže programu přenášejí na druhou *Queue*, kde postupně probíhají následující kroky: Snímání, Čekání, Snímání, Výpočet hodnot pro robota, Čekání pro odeslání do robota, Navázání komunikace s robotem a Odesání.

Snímání využívá funkci *Vision Acquisition*, která dokáže vyhledat dostupnou kameru a zajistí veškerou potřebnou komunikaci. Se snímkem, který je touto cestou zaznamenán, lze dále vhodně pracovat. Na obrázku můžete vidět, jak tato funkce vypadá a jaké je její nastavení. Nejpodstatnější záložka je hned ta první, kde je potřeba nalézt a zvolit potřebnou kameru. V dalších záložkách je pak možno zvolit, zda jako výstup požadujeme jeden snímek, sekvenci snímků či video a je též možné upravit základní nastavení kamery jako je gama, sytost, hloubka, kontrast nebo vyvážení bílé barvy.

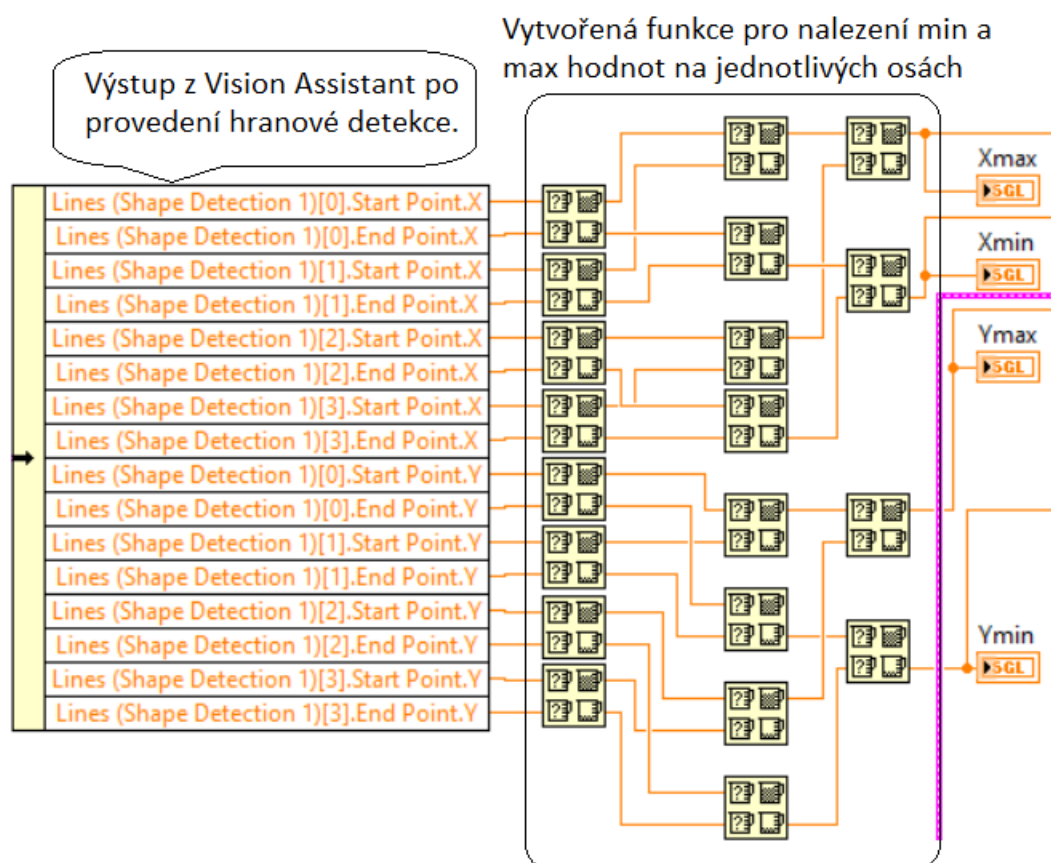


Obrázek č. 17. Funkce Vision Acquisition a její nastavení

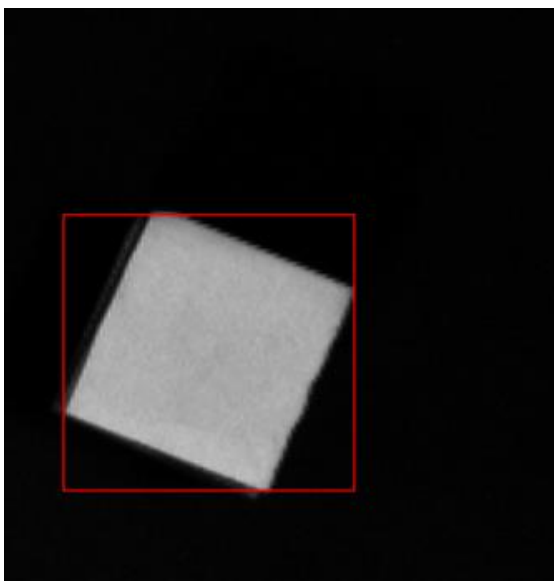
Na výstup z předchozí funkce je možno aplikovat základní filtraci snímku. Vzhledem k tomu, že objekt, který se vyhodnocuje, je barevně kontrastní oproti černému dopravníkovému pásu, jevila se jako nejlepší varianta provést tuto filtraci na základě intenzity světla v sejmutém obrazu. Dalšími možnostmi jsou například, hloubka, či jedna z barev R/G/B. Poté již následuje samotné zpracování snímku, k čemuž je použita opět funkce programu LabView – v tomto případě *Vision Assistant*, jež umožňuje účinnější a rozsáhlejší úpravu snímku s ohledem na snímávaný objekt. Současně je možné provést samotné vyhodnocení, které umožňuje tento programový blok. Neocenitelné pak je především to, že je ihned vidět výsledek, pokud je vyhodnocovaný předmět tou dobou snímán. Také je možné pro nastavování použít jakýkoli uložený obrázek. Možnosti této funkce jsou opravdu velice rozsáhlé, avšak u této aplikace se nejlépe jeví hranová detekce. Při použití hranové detekce jsou výstupem funkce vektory zaznamenaných hran. Jelikož se vyhodnocuje předmět čtvercové podstavy, jsou tyto vektory čtyři. Směry oněch vektorů nejsou nikdy přesně dány a není možné nastavit ani výchozí bod od kterého by měly být udávány, proto je třeba provést dodatečné vyhodnocení.

Použité vyhodnocení vektorů se zakládá na faktu, že je nutné znát pouze střed sledovaného objektu. Pro tento účel byla vytvořena struktura, která vyhodnotí a zaznamená minimální a maximální hodnotu na ose X a na ose Y viz Obrázek č. 18 Vyhodnocení min a max hodnot souřadnic v LabView. Výsledek funkce pro vyhledání minimálních a maximálních

hodnot lze vidět zde: Obrázek č. 19 Výřez snímku z kamery po spočítání min a max na ose X a Y, kde je do obrazu vložen obrys, který je určen min. a max. hodnotami z vyhodnocení. Z nich je možné prostou matematickou funkcí zjistit střed objektu. Zároveň je tímto způsobem možné eliminovat chybu vyhodnocení, jelikož v určitých případech dochází k tomu, že jednotlivé vektory nekorespondují s délkou celé vyhodnocované hrany, ale jsou kratší.



Obrázek č. 18. Vyhodnocení min a max hodnot souřadnic v LabView



*Obrázek č. 19. Výřez snímku z kamery po spočítání min a max na ose  $X$  a  $Y$*

V okamžiku, kdy jsou získány souřadnice středu objektu přejde program na čekací smyčku. Doba strávená v této smyčce je určena konstantou na základě provedeného testování. V případě, že by se do budoucna vyměnila převodovka a dopravník by jezdil rychleji, je možné tuto konstantu upravit přímo z ovládací obrazovky. Po uplynutí čekací konstanty se znovu přejde na snímání a výpočty, kde se zároveň zaznamená, že bylo provedeno čekání a že se jedná o druhé snímání.

Následuje přepočet hodnot. Pro lepší pochopení důvodů a způsobu přepočtu je vložen *Obrázek č. 20* Souřadnice pro přepočet, pod tímto odstavcem. Pro odebrání předmětu robotem je potřeba znát dvě hodnoty. Těmi jsou souřadnice  $X_r$  z pohledu robota a čas, kdy bude předmět v odebírací pozici robota, tj. souřadnice  $Y_r$  z pohledu robota bude rovná nule. Je nutné si uvědomit, že  $S_x$  koresponduje s  $Y_r$ .  $S_y$  pak koresponduje s  $X_r$ , přičemž směry pohybů od nuly do kladných hodnot, jsou u těchto dvou souřadnic opačné.

Nejprve je potřeba vyhodnocené hodnoty  $S_x$  a  $S_y$  převést z pixelů na milimetry, aby byly všechny hodnoty ve stejných jednotkách. Tento převod byl nakonec vypočítán za pomoci snímku z kamery, na kterém byl po celé jeho délce rozvinut metr. Z vyhodnocení v LabView bylo možné odečíst počet pixelů i hodnotu v milimetrech, které bylo třeba mezi sebou pouze podělit, přičemž výsledná hodnota jeden pixel odpovídá 0,4486638537271mm po zaokrouhlení. Toto je nejpřesnější možná hodnota, se kterou jde v programu LabView počítat. Skutečně je potřeba počítat s co nejpřesnějšími čísly, jelikož by i malé zaokrouhlení znamenalo poměrně velkou nepřesnost, kterou si v tomto s ohledem na velikost odebíraného předmětu,



a především vzdálenost kamery od předmětu není možné dovolit. Na snímku dochází ještě k jedné nepřesnosti, která je dána pozicí kamery a polohou předmětu na dopravníku, jelikož předmět není po celou dobu přesně pod středem kamery. K částečné kompenzaci dochází díky dvěma snímkům, kdy je jeden snímek vyhodnocen před tím, než je předmět pod středem kamery a druhý v době kdy je předmět za středem kamery. Pokud by byl předmět zachytáván méně synchronně od středu kamery, byla by nutná kompenzace.

Pro další výpočty je nutné znát rychlost pohybujícího-se objektu. Tu zjistíme za pomoci dráhy, kterou urazil sledovaný předmět po dobu čekací konstanty, tedy použitím známého vzorce:

$$v = \frac{s}{t} \quad (1)$$

V našem případě bude vzorec vypadat následovně:

$$v = \frac{(Sx_1 - Sx_2)}{t} \quad (2)$$

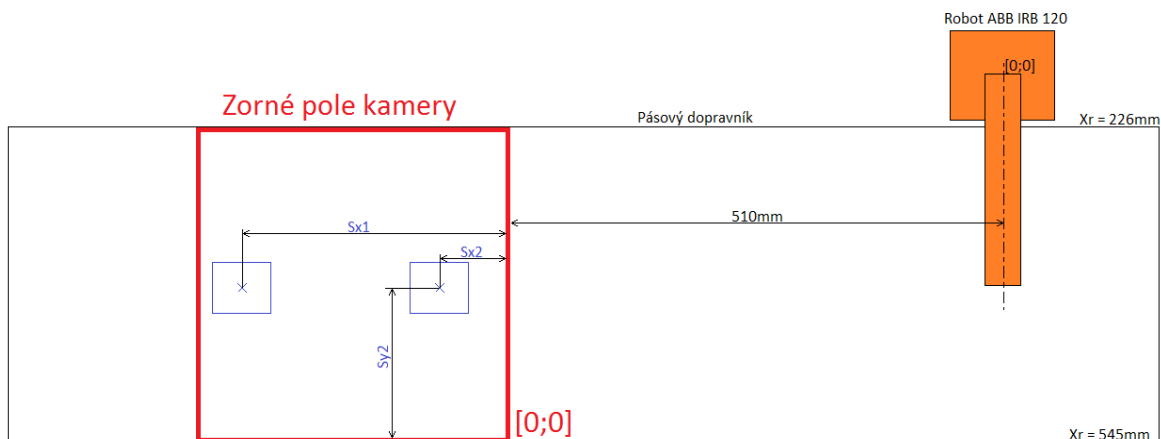
Známe tedy rychlost pohybujícího – se objektu a je možné dopočítat dobu, za kterou bude objekt na souřadnici  $Yr$  rovné nule. K tomu využijeme upravený vzorec číslo 1:

$$t = \frac{s}{v} \quad (3)$$

V tomto případě:

$$t_{celkové} = \frac{Sx_2 + l}{v} \quad (4)$$

Čas je tedy spočítán, zbývá dopočítat souřadnici  $Xr$ . Vychází se ze známé minimální a maximální hodnoty souřadnic pro odběr předmětu a to jak z pohledu kamery, tak i robota. Maximální hodnota  $Xr$  odpovídá minimální hodnotě  $Sy$ , tudíž od maximální hodnoty  $Xr = 545mm$  pouze odečteme skutečně vyhodnocenou hodnotu  $Sy$ .



Obrázek č. 20. Souřadnice pro přepočet

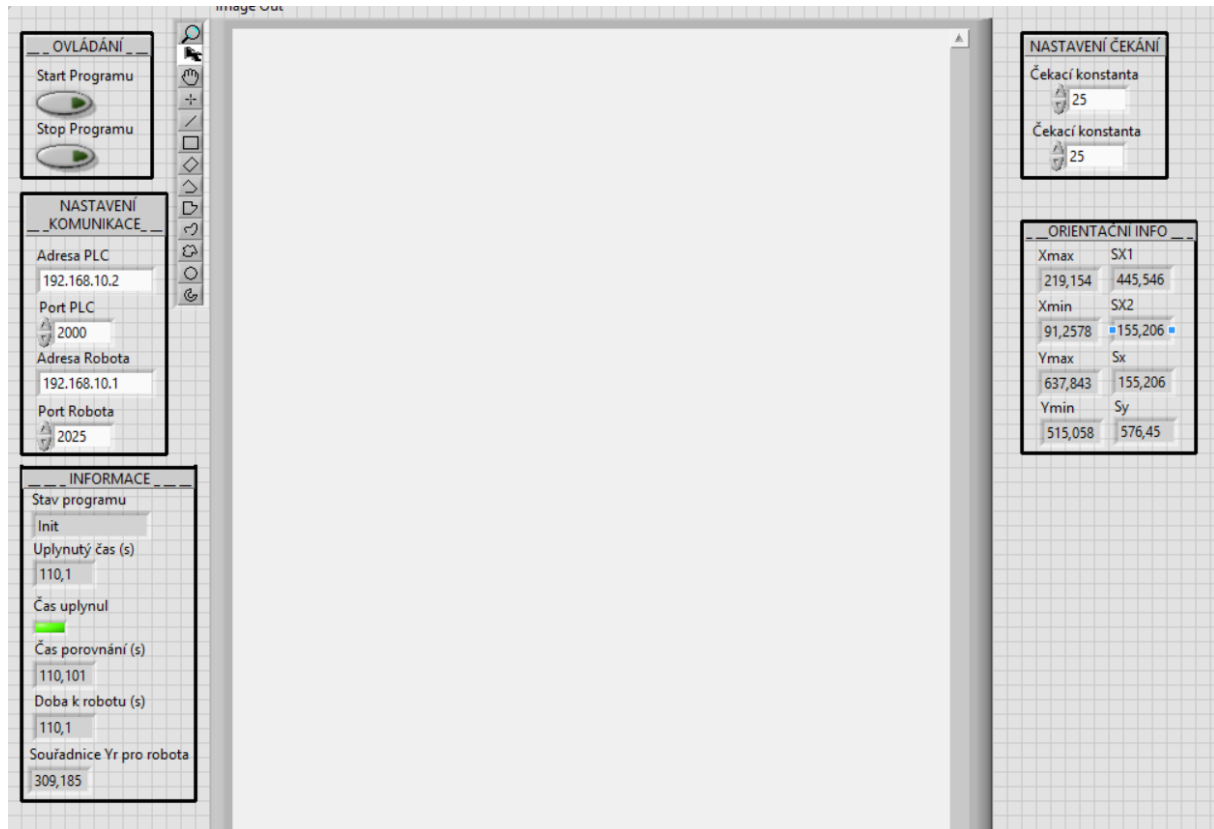
Na základě známé hodnoty času je zahájeno čekání, o které se stará funkce *Elapsed time*, jenž funguje podobně jako stopky. Po uplynutí požadovaného času nastaví výstupní proměnnou na *True*, čehož se dá využít jako impulsu pro odeslání hodnoty souřadnice  $X_r$  do robota.

### 3.3.1. Ovládací rozhraní

Ovládací rozhraní, které je dostupné v programu LabView lze vidět na následujícím obrázku. Je rozděleno na několik kategorií – konkrétně Ovládání, Nastavení komunikace, Informace, Nastavení čekání a Orientační info. Nechybí ani výstupní obraz z kamery včetně vyhodnocených hran.

- Ovládání umožňuje nastartovat a zastavit běh programu.
- Nastavení komunikace je nezbytné pro komunikaci s PLC, kde je výchozí adresa nastavena na 192.168.10.2 a port 2000. Pro komunikaci s řídicím systémem robota je pak adresa 192.168.10.1 a port 2025.
- Informace obsahují informativní text, díky kterému je jasné, ve které části se program za běhu nachází. Dále *čas k porovnání*, který se objeví až po vyhodnocení doby, za kterou se dostane jedoucí předmět na pozici pro odebrání. Zároveň je zobrazován čas, který už z této doby uběhl. Uplynutí času signalizuje indikátor. Položky *Doba k robotu* a *Souřadnice  $Y_r$  pro robota* jsou skutečně vyhodnocené informace.
- V pravé části je nastavení časování, přičemž první hodnota je skutečný čas, po který se čeká mezi jednotlivým snímáním, druhá je použita pro následný výpočet.

- Orientační info vyobrazuje vyhodnocené hodnoty ale v pixelech. Díky tomuto přehledu je možné získat informace o jednotlivých souřadnicích již během vyhodnocování. Především během ladění jsou tyto informace velice užitečné.



Obrázek č. 21. Ovládací rozhraní v LabView

### 3.4. Program robota

V první řadě byl vytvořen model celé situace obsahující robota ABB IRB 120, pásový dopravník, přenášený objekt a nástroj pro robota. Pro možnost použití funkcí *Socket* je nezbytné, aby byl při vytváření virtuálního systému tento systém vytvořen z *Layoutu*, kde máme možnost integrovat *Option PC-Interface*, který tyto funkce podporuje. Dále je potřeba uvážit, zda je nutné vytvářet i *workobject*, či použít výchozí. Pokud jsou různé *robtargety* vztaženy k různým *workobjectům* budou po případném posunutí daného *workobjectu* zachovány pozice *robtargetu*, které jsou k němu vztažené. Vzhledem k jednoduchosti programu nebyl další *workobject* vytvořen.

Nástroj se vytváří stejným způsobem jako například model dopravníku, ale posléze je potřeba ho upravit – určit mu souřadnicový systém hmotnost a síly, které na něj budou působit. Vzhledem k tomu, že je hmotnost a působící síly minimální, byly tyto hodnoty zanedbány. Do

tohoto modelu se implementovala virtuální cesta s pevně danými pozicemi *robtargetů*. Také byla potřeba nadefinovat virtuální jednotku a virtuální signál, aby bylo možné je v simulaci použít – v reálném řídicím systému robota je tato jednotka i signál připravena dle hardwareové karty.

Virtuální jednotku i signál lze přidat v záložce *Controller* zvolením *I/O* v nabídce *Configuration Editor*. Následně se vybere *Unit* a pomocí pravého tlačítka se vytvoří nová jednotka, která bude typu *Virtual1*. Tento typ je již vytvořen v softwaru od ABB.

Type	Name	Type of Unit	Connected to Bus	Unit Identification Label	Unit Trustlevel	Unit Startup State	Store Unit State at Power Fail	Regain Communication Reset
Access Level	DRV_1	LOCAL_GENERIC	Local	D611 Cont. board	Loss accepted (2)	Deactivated	No	Disabled
Bus	DRV_2	LOCAL_GENERIC	Local	D611 Cont. board	Loss accepted (2)	Deactivated	No	Disabled
Cross Connection	DRV_3	LOCAL_GENERIC	Local	D611 Cont. board	Loss accepted (2)	Deactivated	No	Disabled
Fieldbus Command	DRV_4	LOCAL_GENERIC	Local	D611 Cont. board	Loss accepted (2)	Deactivated	No	Disabled
Fieldbus Command Type	PANEL	LOCAL_GENERIC	Local	D630 Panel board	Loss accepted (2)	Activated	No	Disabled
Route	Virtual	Virtual	Virtual1	1	Error when lost (1)	Activated	No	Disabled

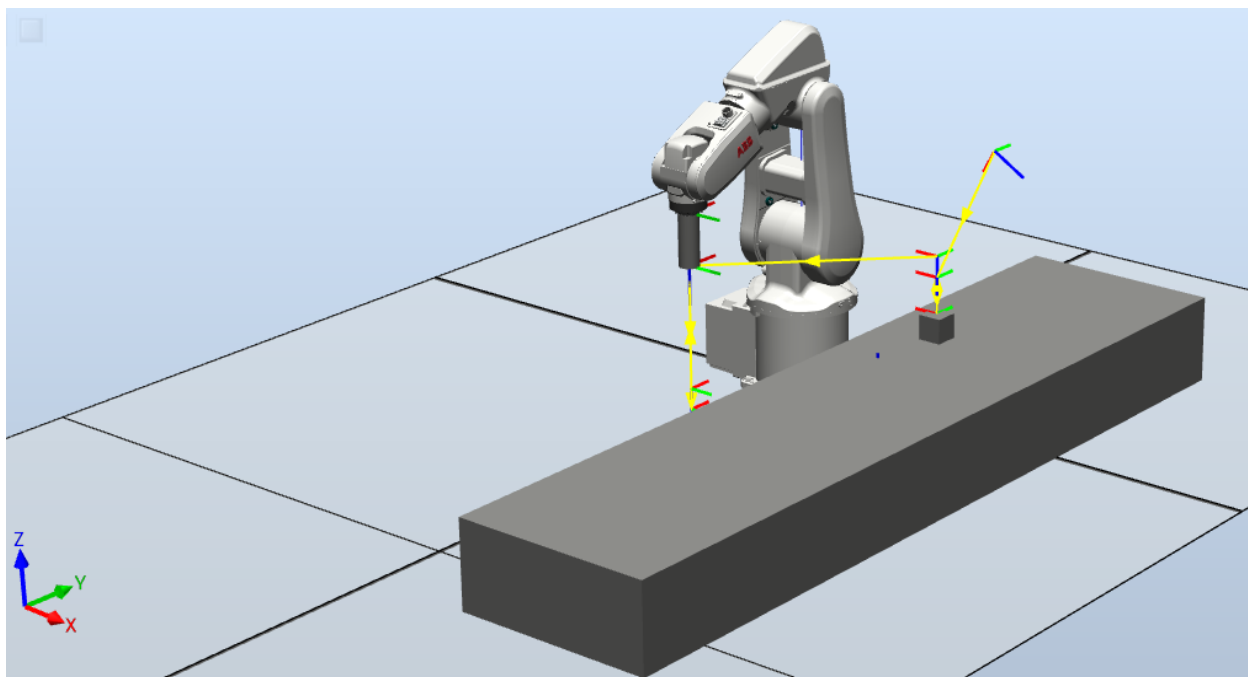
Obrázek č. 22. Vytvoření virtuální jednotky

Virtuální signál, se pak vytváří na téže záložce, akorát se v levém sloupci vybere *Signal*. Tento signál pak musí být přiřazen k vytvořené virtuální jednotce a je třeba jej nastavit, dle typu požadovaného signálu – v tomto případě digitální výstup. Vytvoření virtuálního signálu je nezbytné, jestliže chceme v simulaci přichytávat a přenášet objekty pomocí nástroje. V novějších verzích RobotStudia je tento postup značně zjednodušen, kdy není potřeba vytvářet virtuální jednotku, ale stačí pouze vytvořit signál, kterému se nepřihadí žádná reálná jednotka. Tento signál je pak automaticky chápán jako virtuální.

Type	Name	Type of Signal	Assigned to Unit	Signal Identification Label	Unit Mapping	Category	Access Level	Default Value	Signal Value at System Failure and Power Fail
Access Level	VirtualDO	Digital Output	Virtual		1		Default	0	Keep Current Value (no change)
Bus	USERDOOVL	Digital Input	PANEL	Overload of user DO	11	safety	INTERNAL	0	N/A
Cross Connection	TESTEN1	Digital Output	PANEL	Activate Glitchtest for Enable1	6	safety	INTERNAL	0	Keep Current Value (no change)
Fieldbus Command	STLEDREDBNK	Digital Output	PANEL	Set status LED to red flashing	8	safety	INTERNAL	0	Keep Current Value (no change)
Fieldbus Command Type	STLEDRED	Digital Output	PANEL	Set status LED to red color	7	safety	INTERNAL	0	Keep Current Value (no change)
Route	STLEDGRBNK	Digital Output	PANEL	Set status LED to green flashing	10	safety	INTERNAL	0	Keep Current Value (no change)
Signal	STLEDGR	Digital Output	PANEL	Set status LED to green color	9	safety	INTERNAL	0	Keep Current Value (no change)
System Input	SS2	Digital Input	PANEL	Superior Stop chain backup(X6.5 to X6.6) and (X6.3 to X6.1)	20	safety	INTERNAL	0	N/A
System Output	SS1	Digital Input	PANEL	Superior Stop chain(X6.4 to X6.6) and (X6.2 to X6.1)	19	safety	INTERNAL	0	N/A
Unit	SOFTSSO	Digital Output	PANEL	Soft Superior Stop	3	safety	INTERNAL	0	Keep Current Value (no change)
Unit Type	SOFTSSI	Digital Input	PANEL	Soft Superior Stop	21	safety	INTERNAL	0	N/A
	SOFTGSO	Digital Output	PANEL	Soft General Stop	2	safety	INTERNAL	0	Keep Current Value (no change)

Obrázek č. 23. Vytvořený virtuální signál

Pokud jsou tedy připraveny všechny potřebné signály a je vytvořen layout celé situace, lze vytvořit základní trajektorii, kterou si posléze pomocí synchronizace převedeme do programu *Rapid* pro lepší možnost upravování potřebných instrukcí.



Obrázek č. 24. Model programu robota

Po vytvoření této základní trajektorie se nadefinovaly proměnné *robtargety* tak, aby bylo možné na základě příchozí komunikace měnit hodnotu jejich souřadnice na ose X, nejsou tedy definovány jako *CONST*, ale jako *VAR*. Pro komunikaci s řídicím systémem robota, tedy i zaslání souřadnice pro odběr předmětu je využit TCP/IP standard, pro jehož využití v reálném řídicím systému robota, je zapotřebí správně nastavený a nahráný systém robota a též *Option PC-Interface*, který není samozřejmostí. Způsob vytvoření a nahrání systému do robota a problémy s tímto spjaté jsou podrobněji vysvětleny v kapitolách: 3.5 Vytvoření a nahrání systému robota, 4.3 Vytvoření systému robota a 4.4 Nahrání systému robota. Po korektním vytvoření a nastavení systému robota je možné používat funkce *Socket*. Vzhledem k tomu, že je PLC pro komunikaci s LabView použito jako server, je i robot použit jako server. Toto umožňuje funkce *SocketBind*, kde je nadefinována IP adresa a port pro příchozí komunikaci, *SocketAccept*, díky níž se příchozí komunikace přijme a *SocketReceive*, která obdrženou informaci zapíše do proměnné. Celá komunikace je pak vidět na následujícím obrázku.

```
SocketClose server_socket;
SocketCreate server_socket;
SocketBind server_socket, "192.168.10.1", 2025;
SocketListen server_socket;
WHILE TRUE DO
    SocketAccept server_socket, client_socket;
    SocketReceive client_socket \Str := str_data;
    SocketClose client_socket;
```

Obrázek č. 25. Komunikace na straně robota

Hodnota, která je přijata a uložena jako *string*, je následně přetypována na číslo a je ověřeno, zda přetypování proběhlo správně – *ok* je nastaveno na *True*. Poté se tato hodnota porovná s vymezeným rozsahem pro maximální a minimální meze souřadnice X a následně se použije pro změnu pozice *robtargetu*.

```
ok:=StrToVal(str_data, X);  
IF (ok = TRUE) THEN  
    IF ((X > 225) AND (X < 546)) THEN
```

Obrázek č. 26. Přetypování a ověření příchozí hodnoty

Proměnné *robtargety* T20, T30 a T40 si nejprve načtou fixní hodnoty s konstantích *robtargetů* Target\_20, Target\_30 a Target\_40 aby byla těmto hodnotám pomocí příkazu *.trans.x*, upravena souřadnice X, ve které je již přetypovaná hodnota poslaná z LabView.

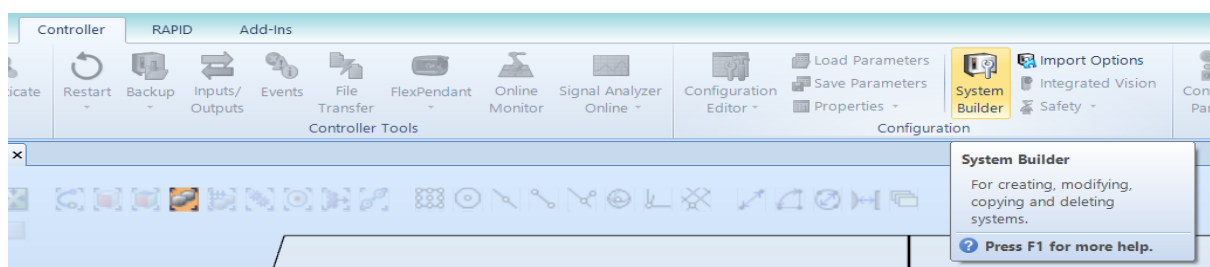
```
T20:=Target_20;  
T20.trans.x:=X;  
T30:=Target_30;  
T30.trans.x:=X;  
T40:=Target_40;  
T40.trans.x:=X;
```

Obrázek č. 27. Přirazení poslané hodnoty *robtargetu*

Pro odebírání předmětu je nutné nastavovat digitální výstup, který ovládá signál přívodu vzduchu do použitého pracovního nástroje na šesté ose robota. V simulaci se pro tyto účely vytváří virtuální jednotka a virtuální signál, kdežto v reálném systému se jednotka i signál musí definovat podle konkrétních vstupů a výstupů na kartě řídicího systému robota.

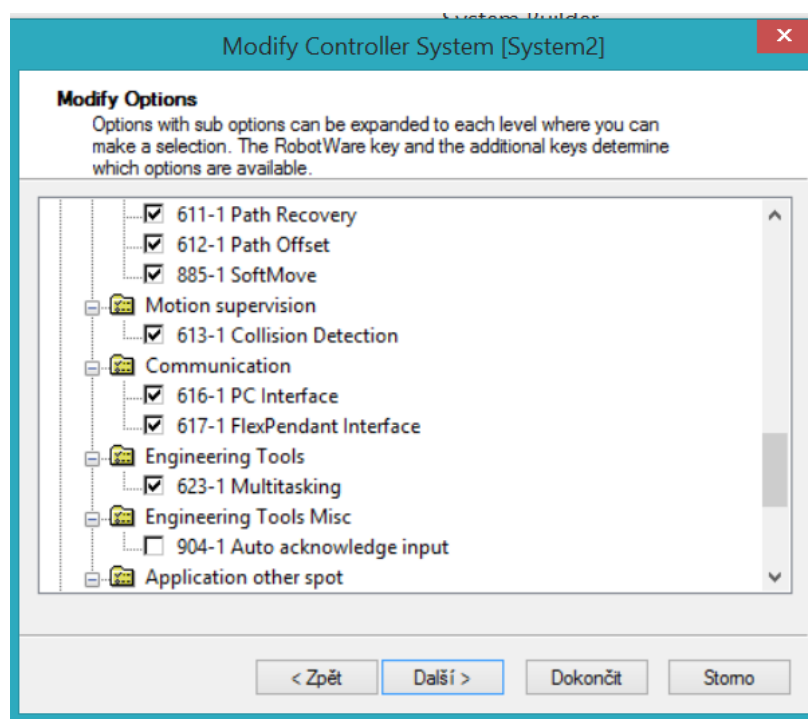
### 3.5. Vytvoření a nahrání systému robota

Vzhledem k tomu, že stávající systém, který byl nahraný v řídicím systému robota neobsahoval *Option PC-Interface*, bylo potřeba vytvořit nový systém a nahrát ho do řídicího systému robota. Nový systém se v RobotStudios vytváří na záložce *Controller*, pomocí ikony *System Builder*.



Obrázek č. 28. System builder

Pro vytvoření je nutné znát *Controller Key*. Zároveň je nezbytné, aby byl nainstalován RobotWare, který obsahuje správné *MediaPool*. Další nutnou položkou, pro vytvoření systému je *Drive Modul Key*, který musí korespondovat s *Controller Key*. Bez této shody není možné reálný systém vytvořit. Poté je již možné vybrat v položce *Options* konkrétní specifikace od jazykové mutace, přes hardware a pohybové funkce až po funkce komunikační, které zahrnují právě *PC-Interface*.



Obrázek č. 29. Možnost výběru Options

Nahrání systému do řídicího systému robota je možné dvěma způsoby, a to pomocí Flash disku nebo přímým spojením s řídicím systémem robota pomocí Service portu. Po nahrání je možné změnit IP adresu, na které má řídicí systém komunikovat, což bylo v tomto případě nezbytné. V případě potřeby je možné používat i původní systém, stačí zvolit restart s bootováním systému a vybrat z nahrených systémů ten požadovaný.

### 3.6. Odebírání předmětu

Na přiložených videích jsou vidět dvě možné varianty pro odebrání. Jednou možností je využití chapadla v případě, že je ve středu pohybujícího-se předmětu umístěn například váleček. Chapadlo je schopno tento váleček zachytit. Druhou možností je využití přísavky, která je schopna bez potíží odebrat předmět čtvercové podstavy. V programu robota je pak nepatrný rozdíl v ose  $z$ , která je v případě použití přísavky o jeden centimetr níže vlivem větší konstrukce nástroje.



## 4. Problémy

### 4.1. Převodovka

Vzhledem k tomu, že má převodovka, která byla k dispozici, moc velký převodový poměr pohybuje se dopravníkový pás velice pomalu i přes maximální možnou nastavenou rychlost z frekvenčního měniče. Bohužel nelze tento problém odstranit jinak, než výměnou převodovky což bylo s ohledem na finanční náklady vyloučeno a rychlost dopravníkového pásu zůstala nízká.

### 4.2. Zpožděné zpracování hran

Ve výjimečných případech nestihl program v LabView vlivem většího množství výpočtů a náročnosti pro zpracování vyhodnotit druhý snímek v době, kdy je ještě celým objemem v zorném poli kamery. Avšak tento problém byl vyřešen zkrácením doby čekání mezi dvěma vyhodnoceními.

### 4.3. Vytvoření systému robota

V tomto případě bylo potřeba vytvořit celý systém kompletně znovu. Starší verze RobotWare i řídicí systém robota bohužel neumožňují dělat takovéto zásahy ve stávajícím systému. V dostupném Robotstudiu – verzi 5.61 se stejnou verzí RobotWare systém také nelze vytvořit. RobotWare jde nainstalovat odděleně od RobotStudia, avšak při samostatné instalaci RobotWare se soubory ve složce *mediapool* nenainstalují korektně. Nainstalují se pouze jako virtuální a pro reálný systém je není možno využít. Pro správnou funkci je lepší použít starší RobotWare, třeba verzi 5.15.13, ale je potřeba tuto verzi stáhnout poněkud komplikovaným postupem. Tento postup není veřejně dohledatelný a je následující: Nainstalovat Robotstudio verze 6.01 a vyšší, kde je možné i na nezalicensované verzi přes záložku add-ins stáhnout instalační soubor pro jakýkoli RobotWare. Po nainstalování RobotWare staženého tímto způsobem jsou již soubory ve složce *mediapool* správné, a je tak možné vytvořit systém.

### 4.4. Nahrání systému robota

I tento krok by neměl být problémový, avšak skutečnost je poněkud odlišná. Oficiálně existují dva postupy, jak nahrát vytvořený systém do řídicího systému robota. Jedním z nich je

překopírovat vytvořený systém na Flash disk, avšak tento Flash disk musí být naformátován jako FAT32 – formát NTFS FlexPendant nedokáže načíst. Další podmínkou je, že na Flash disku je pouze systém, jehož nahrání je požadováno, nicméně po naformátování na FAT32 zůstávají na Flash disku určité soubory, které brání v nahrání. Tento postup tedy použít nejde a je potřeba systém nahrát přes počítač za pomoci servisního portu řídicího systému robota.

## 4.5. Zaznamenání objektu na dopravníkovém pásu

Prvním zadrhelem, který nastal byla výška sledovaného objektu. Standardní kostka, která se používá pro podobné aplikace v laboratoři robotiky na TUL je relativně malá a senzor polohy ji nezaznamená. Proto byly v prvním případě využity dvě kostky na sobě. Později bylo převrtáno uchycení jednotlivých senzorů polohy. Díky této úpravě stačí použít standardně jedna kostka.

Dalším problémem je ne zcela dostatečné osvětlení, ač jsou nad dopravníkem čtyři výkonné halogeny, objekt není dostatečně světlý, aby ho kamera zaznamenala, proto bylo potřeba snímanou plochu zesvětlit. Čím je předmět světlejší, tím je snímání lepší a efektivnější.

## 5. Závěr

Cílem práce bylo vytvoření funkčního robotického pracoviště, využívajícího několik technických oblastí současně. Tento cíl byl naplněn a jeho výsledek můžete shlédnout na přiloženém videu. Úloha funguje komplexně, ale je možné ji využívat i jako dvě oddělené části.

První část tvoří pásový dopravník, jehož pohyb je řízen pomocí PLC firmy B&R. Pohyb předmětu na dopravníku je snímán za pomoci infračervených senzorů polohy. Na základě informací, které jsou z nich získávány PLC nastavuje základní rychlost a směr pohybu dopravníku, který zajišťuje frekvenční měnič a asynchronní motor. Při nákupu byla bohužel zvolena převodovka s horšími parametry, a tudíž je rychlost dopravníku poměrně nízká, což by byl v běžném provozu značný problém. Vzhledem k tomu, že na Technické univerzitě v Liberci bude dopravník využíván pouze během výuky či dnů otevřených dveří, můžeme si dovolit motor více přetěžovat a trochu tím zkrátit jeho životnost. Dalším řešením by byla výměna převodovky, ale zkrácení životnosti bude v tomto provozu sotva znatelné, proto se jeví jako přijatelnější varianta řešení.

Druhou, samostatnou část tvoří – v laboratoři umístěný – průmyslový robot firmy ABB, konkrétně ABB IRB 120, který spolupracuje s inteligentní kamerou umístěnou nad pásovým dopravníkem. Na základě kamerou snímaného obrazu a následného vyhodnocení za pomoci software LabView je robot schopen odebírat statické, či pohybující-se předměty. K jejich odběru může využít přísavky, ale je možné použít i kleště za předpokladu, že na středu snímaného objektu bude umístěn dostatečně malý předmět. V případě pohybujícího se předmětu software v LabView dopočítává rozdíl vzdáleností, který nastane v čase, jež předmět urazí během vyhodnocení kamerou a najetím robota do odebírané polohy, aby byl předmět zachytáván, pokud možno, v jeho středu.

Obě tyto části současně tvoří funkční celek. Přibližovací senzory na dopravníku zachytí přibližující se objekt, načež zareaguje PLC, které vyhodnotí pozici a natočení objektu. Na základě získaných souřadnic odebere průmyslový robot pohybující se objekt a přemístí jej na požadované místo. Proces odebrání je vidět na přiložených videích.

Návrh a realizace tohoto řešení se neobešel bez komplikací. První bylo řešení samotného dopravníkového pásu. Ideální by bylo mít k dispozici článkový dopravník, který by tvořil okruh, nicméně tyto dopravníky jsou desetinásobně dražší a z toho důvodu byla zvolena varianta s koncovými senzory polohy a otáčením směru pohybu. Dalším zádrhelem byla

převodovka – tento problém byl objasněn ve druhém odstavci této kapitoly. Problematika dobrého osvětlení, které potřebuje inteligentní kamera ke správné funkčnosti, byla v laboratoři vyřešena již dříve, bohužel pro tuto aplikaci nedostatečně. Nejjednodušší řešení je použít světlejší objekt, který se odebírá.

Vhodným a zajímavým vylepšením celé aplikace by bylo například rozšíření o spolupráci s dalším robotem, tedy vzájemná spolupráce dvou robotů nebo třeba vyskládání odebraných objektů do konkrétního útvaru. Zajímavá by byla i možnost měnit pomocí jedné proměnné nastavení robota, na jejímž základě by robot mohl pracovat buď s přísavnou nebo s chapadlem.

Pro lepší využití samostatného dopravníku by bylo vhodné rozšířit PLC o panel s vizualizací, aby odpadla potřeba používání PC k jeho jednoduchému chodu. Celá úloha může ke spolupráci využívat několika robotů a utvářet tak kompletní pracoviště. Využívá robotiku, počítačové vidění, zpracování obrazu, automatizaci a elektroniku – evokuje tak reálně vytvářené aplikace v průmyslu a poskytuje velký potenciál a možnosti při výuce na Technické univerzitě v Liberci.

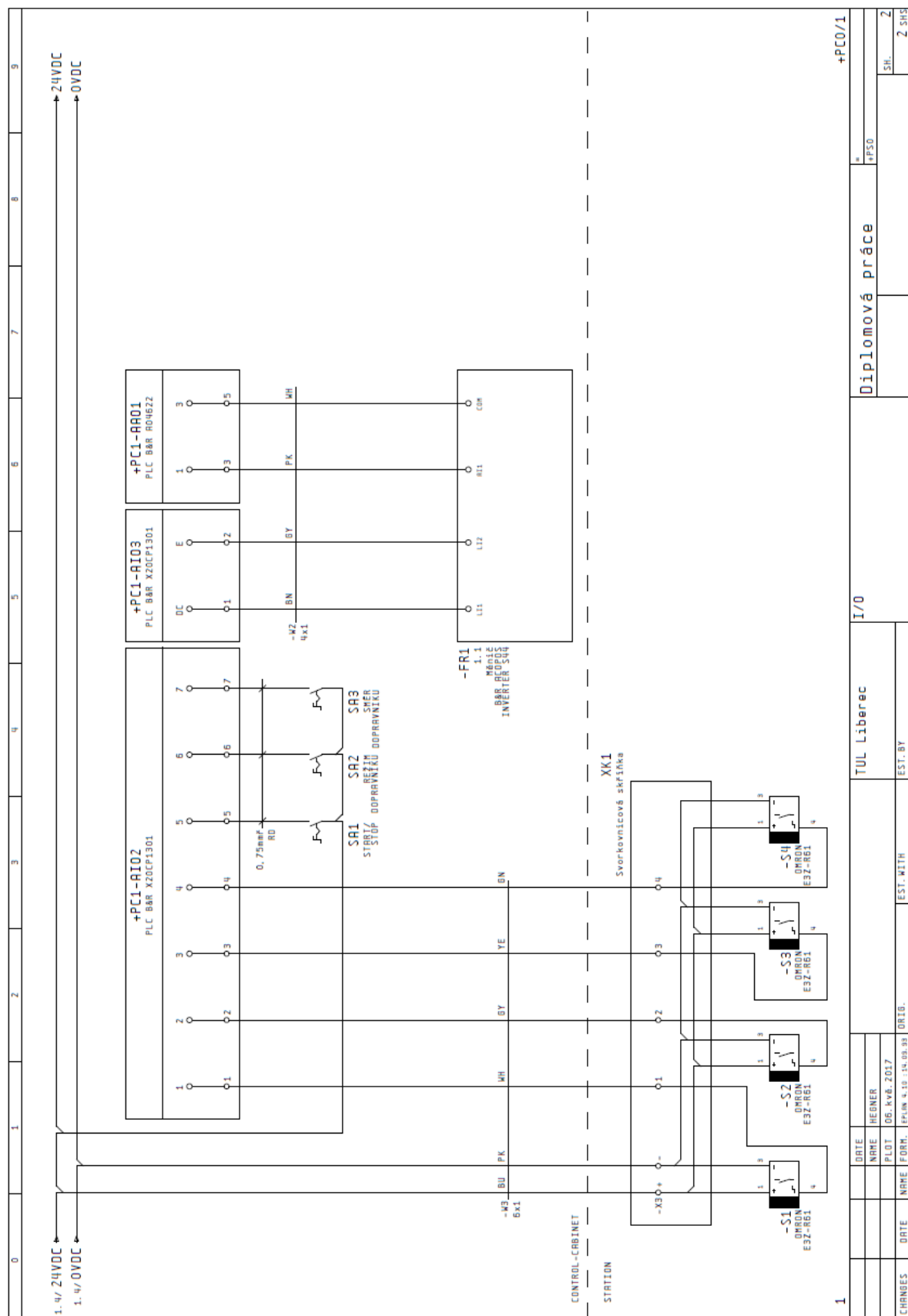
# Citace

- [1] Craig J., J.: Introduction to Robotics Mechanics and Control, Third Edition, Prentice Hall, 2005.
- [2] Delores M. Etter and Jeanine A. Ingber.: Engineering Problem Solving with C++, 3rd edition, Prentice Hall Press, Upper Saddle River, NJ, USA, 2011.
- [3] Dietrich W. R. Paulus, Joachim Hornegger: Pattern Recognition and Image Processing in C++, 1st edition, Verlag Vieweg, Braunschweig/Wiesbaden, 1995.
- [4] Dopravníky [online]. 2012 [cit. 08.05.2017]. Dostupné z: [http://mechmes.websnadno.cz/dokumenty/pri-str-01.07\\_dopravniky.pdf](http://mechmes.websnadno.cz/dokumenty/pri-str-01.07_dopravniky.pdf)
- [5] Obrázek. Dostupné z: [http://www.zivefirmy.cz/media/fotos/1017152/engel-strojirenska\\_max57697.jpg](http://www.zivefirmy.cz/media/fotos/1017152/engel-strojirenska_max57697.jpg)
- [6] Obrázek. Dostupné z: <http://www.askmt.com/obrazek-embs-clankove-dopravniky-neprime-404>
- [7] NORD. Datasheet G1000\_CZ\_1810.pdf [online]. [cit. 08.05.2017]. Dostupné z: [https://www.nord.com/cms/media/documents/bw/G1000\\_CZ\\_1810.pdf](https://www.nord.com/cms/media/documents/bw/G1000_CZ_1810.pdf)
- [8] Obrázek. Dostupné z: <https://www.ebay.com/p/NORD-SK-63-S-4-CUS-Bre5-HL-230-460v-3-phase-Motor-W-40-1-Gearbox/666524812>
- [9] B&R Automation. Datasheet X20(c)CP13xx [online]. 2017 [cit. 08.05.2017]. Dostupné z: <https://www.br-automation.com/cs/soubory-ke-stazeni/#categories=control-and-io-systems/x20-system/cpus/x20cp1381>
- [10] Obrázek. Dostupné z: [http://elmecon-mk.com/wpcontent/uploads/2014/01/e3z\\_omron.jpg](http://elmecon-mk.com/wpcontent/uploads/2014/01/e3z_omron.jpg)
- [11] Omron. Datasheet e3z\_ds\_e\_18\_2\_csm438.pdf [online]. 2015 [cit. 08.05.2017]. Dostupné z: [https://www.ia.omron.com/data\\_pdf/cat/e3z\\_ds\\_e\\_18\\_2\\_csm438.pdf?id=407](https://www.ia.omron.com/data_pdf/cat/e3z_ds_e_18_2_csm438.pdf?id=407)
- [12] Obrázek. Dostupné z: <http://www.prumyslove-kamery.cz/o-produktech/digitalni-kamery-basler/plosne-maticove-kamery/ace>
- [13] Basler Ace. [online].2012 [cit. 08.05.2017]. Dostupné z: <http://www.aegis-elec.com/basler-aca1300-30gm-gc-area-scan-gige-cameras.html>
- [14] CCD kamery pro astronomii. [online].2011 [cit. 08.05.2017]. Dostupné z: <http://www.gxccd.com/art?id=303&lang=405>

- [15] Pavla Žbáňková, Projekt ke zkoušce ze Základů obrazového inženýrství - CCD senzory, [online]. 2002. [cit. 08.05.2017]. Dostupné z: [http://www.fch.vut.cz/~zmeskal/obring/presentace\\_2002/07\\_ccd\\_senzory.pdf](http://www.fch.vut.cz/~zmeskal/obring/presentace_2002/07_ccd_senzory.pdf)
- [16] Obrázek. Dostupné z: <https://answerszips.jimdo.com/2016/01/24/national-instruments-gige-vision-driver/>
- [17] Petr Polák. Diplomová práce – Kontrola stability sítě v reálném čase pomocí přístroje Leica TCA2003. Prosinec 2007. ČVUT, Praha.
- [18] TCP/IP. Pedagogická fakulta Masarykovy univerzity [online]. [cit. 08.05.2017]. Dostupné z: [http://www.ped.muni.cz/wtech/03\\_studium/teps/teps-03.pdf](http://www.ped.muni.cz/wtech/03_studium/teps/teps-03.pdf)
- [19] ABB S4C [online]. [cit. 08.05.2017]. Dostupné z: <http://www.used-robots.com/abb/controller/s4c>
- [20] ABB S4Cplus *data sheet* [online]. 2000 [cit. 08.05.2017]. Dostupné z: [http://www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/\\$FILE/Datasheet%20S4Cplus.pdf](http://www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/$FILE/Datasheet%20S4Cplus.pdf)
- [21] ABB IRC5 *data sheet* [online]. 2011 [cit. 08.05.2017]. Dostupné z: [http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/c13e1c5490c61230c125796000515137/\\$file/IRC5%20datasheet%20PR10258%20EN\\_R13.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/c13e1c5490c61230c125796000515137/$file/IRC5%20datasheet%20PR10258%20EN_R13.pdf)
- [22] Jakub Hegner. Bakalářská práce – Demoúlohy průmyslových robotů ABB. 2014. Technická Univerzita v Liberci.
- [23] ABB. IRB140 *data sheet* [online]. 2003 [cit. 08.05.2017]. Dostupné z: [http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/98ba43a906331fec48257c6f00374818/\\$file/PR10031EN%20R15\\_En.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/98ba43a906331fec48257c6f00374818/$file/PR10031EN%20R15_En.pdf)
- [24] ABB. IRB1400 *data sheet* [online]. 2005 [cit. 08.05.2017]. Dostupné z: [http://www05.abb.com/global/scot/scot230.nsf/veritydisplay/99bb3fb8ff6495cfc1257b130056d120/\\$file/IRB1400\\_R3-US%2002\\_05.pdf](http://www05.abb.com/global/scot/scot230.nsf/veritydisplay/99bb3fb8ff6495cfc1257b130056d120/$file/IRB1400_R3-US%2002_05.pdf)
- [25] ABB. IRB120 *data sheet* [online]. 2012 [cit. 08.05.2017]. Dostupné z: [http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/\\$file/ROB0149EN\\_D\\_LR.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/$file/ROB0149EN_D_LR.pdf)
- [26] ICX445AKA. Datasheets360. [online]. [cit. 08.05.2017]. Dostupné z: <http://www.datasheets360.com/pdf/461808205957723071>
- [27] Aleš Kepřt. Komunikační sokety [online]. 2006. UP Olomouc. [cit. 08.05.2017]. Dostupné z: <http://www.keprt.cz/texty/sockety.pdf>

## Elektrické zapojení rozvaděče a PLC





Obrázek č. 31. Elektrické zapojení 24V



## Obsah CD

- Text diplomové práce
  - Diplomová\_Práce\_2017\_Jakub\_Hegner.docx
  - Diplomová\_Práce\_2017\_Jakub\_Hegner.pdf
  - Abstrakt\_Diplomová\_Práce\_2017\_Jakub\_Hegner.docx
  - Abstrakt\_Diplomová\_Práce\_2017\_Jakub\_Hegner.pdf
  - Kopie\_Zadání\_Diplomové\_Práce\_Str1.png
  - Kopie\_Zadání\_Diplomové\_Práce\_Str2.png
- Video
  - Odebrání\_Kostky\_1.mov
  - Odebrání\_Kostky\_2.mov
  - Odebrání\_Válečku\_1.mov
  - Odebrání\_Válečku\_2.mov
- Zdrojové kódy
  - LabView
  - PLC\_B&R
  - RobotStudio
- Elektrické zapojení rozvaděče
  - Elektrické zapojení rozvaděče.pdf